

Auslegung und Bahnplanung einer 8-Achsen Parallel-Seriell Bewegungsplattform

Design and Path Planning of an 8-Axis Parallel-Serial Motion Platform

Wissenschaftliche Arbeit zur Erlangung des Grades
Master of Science (M.Sc.)

an der Fakultät für Elektrotechnik und Informationstechnik der Technischen
Universität München.

Themenstellender Prof. Dr.-Ing. Martin Otter
Lehrstuhl für Elektrische Antriebssysteme und Leistungselektronik

Betreuer Dipl.-Ing. Andreas Seefried
DLR - Institut für Systemdynamik und Regelungstechnik

Eingereicht von Thomas Bernhofer

Eingereicht am 31.08.2021 in Garching b. München

Abstract

In motion simulation, motion platforms enable the simulation of forces and angular velocities acting on a person in a car or in an airplane, for example. Other applications for motion platforms are, for instance, perceptual research or testing the dynamic behavior of mechanical structures. In this work, a joint path planning module for a novel motion platform has been developed, which allows it to follow a cartesian trajectory. The motion platform consists of a hexapod on which two additional rotational axes are mounted. The joint path planning module is designed based on the inverse kinematics of the motion platform, which was due to the hybrid and redundant structure calculated using differential kinematics. Therefore, multiple tasks with different priorities can be defined directly considering the position constraints of the hexapod's actuators. Additionally, the velocity and acceleration of the actuators are constrained. A multibody simulation model was used to verify the joint path planning. It was found that the joint path planning works, but it should be noted that the integration of the position limits in the inverse kinematics results in velocity jumps, which can be problematic for motion simulation. Another objective of this work was to design the drives for the two additional axes. By solving an optimization problem, joint trajectories were determined in which the drive torques become maximum. With the help of these torques, the drives for axis 7 and axis 8 can be designed.

Zusammenfassung

In der Bewegungssimulation ermöglichen Bewegungsplattformen die Erzeugung von Kräften und Drehraten, die beispielsweise in einem Auto oder in einem Flugzeug auf eine Person einwirken. Weitere Einsatzbereiche von Bewegungsplattformen stellen zum Beispiel die Wahrnehmungsforschung oder das Testen des dynamischen Verhaltens von mechanischen Strukturen dar. In dieser Arbeit wurde ein Gelenkbahnplanungsmodul für eine neuartige Bewegungsplattform entwickelt, mit dem es möglich ist kartesische Trajektorie abzufahren. Die Bewegungsplattform besteht aus einem Hexapod, auf dem zwei zusätzliche rotatorische Achsen montiert sind. Die Gelenkbahnplanung wurde aufbauend auf der inversen Kinematik der Bewegungsplattform erstellt, welche aufgrund der hybriden und redundanten Struktur mithilfe der differentiellen Kinematik berechnet wurde. Dabei wurde ein Verfahren eingesetzt, bei dem mehrere Aufgaben mit unterschiedlichen Prioritäten unter direkter Berücksichtigung der Positionsbeschränkungen der Aktuatoren des Hexapods definiert werden können. Zusätzlich wurden auch die Geschwindigkeit und Beschleunigung der Aktuatoren beschränkt. Um die Gelenkbahnplanung zu verifizieren, wurde ein Mehrkörpersimulationsmodell eingesetzt. Dabei wurde festgestellt, dass die Gelenkbahnplanung funktioniert, es ist jedoch festzuhalten, dass durch die Integration der Positionslimits in der inversen Kinematik Geschwindigkeitssprünge entstehen, welche für die Bewegungssimulation problematisch sein können. Ein weiteres Ziel dieser Arbeit war es die Antriebe für die beiden zusätzlichen Achsen auszulegen. Hierbei wurden durch das Lösen eines Optimierungsproblems Gelenktrajektorien ermittelt, bei denen die Antriebsmomente maximal werden. Mithilfe dieser Momente können die Antriebe für Achse 7 und Achse 8 ausgelegt werden.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen zu Bewegungsplattformen	3
2.1	Strukturen von Bewegungsplattformen	3
2.2	Steueralgorithmen für Bewegungsplattformen	7
2.3	8-Achsen Bewegungsplattform	8
2.3.1	Systembeschreibung	8
2.3.2	Vor- und Nachteile der Bewegungsplattform	8
3	Kinematische Modellbildung	10
3.1	Kinematische Grundlagen	10
3.2	Serielle Struktur	14
3.2.1	Direkte Kinematik	14
3.2.2	Inverse Kinematik	17
3.3	Parallele Struktur	18
3.3.1	Inverse Kinematik	19
3.3.2	Direkte Kinematik	21
3.4	Hybride Struktur	22
3.4.1	Direkte Kinematik	22
3.4.2	Inverse Kinematik	23
3.5	Inverse Kinematik mit Prioritäten	25
3.6	Inverse Kinematik mit Rückführung	27
4	Gelenkbahnplanung	30
4.1	Aktuatorbeschränkungen	31
4.2	Definition der Aufgaben	32
4.3	Funktionsweise der Gelenkbahnplanung	34
5	Implementierung	38
5.1	Mehrkörpersimulationsmodell	38
5.2	Visualisierung	41
5.3	Gelenkbahnplanungsmodul	41
6	Verifizierung der Gelenkbahnplanung	45
6.1	Erklärung der vorgegebenen Trajektorie	45
6.2	Parametrierung der Gelenkbahnplanung	45
6.3	Resultierende Simulatorbewegungen und Analyse	47
7	Antriebsauslegung	51
7.1	Berechnung der maximalen Antriebsmomente	51
8	Zusammenfassung und Ausblick	56
A	Elementardrehungen und Einheitsvektoren	63
B	Weitere Simulationsergebnisse der Gelenkbahnplanung	64
C	Weitere Simulationsergebnisse der Antriebsauslegung	67

1. Einleitung

Bewegungsplattformen werden in der Flug- und Fahrsimulation eingesetzt, um ein reales Bewegungsgefühl zu imitieren. Dabei werden die Kräfte und Drehraten, welche beispielsweise in einem Auto oder in einem Flugzeug auf eine Person einwirken, reproduziert. Sie spielen daher eine wichtige Rolle in der Pilotenausbildung und in der Fahrzeugentwicklung. Außerdem werden Bewegungsplattformen in der Wahrnehmungsforschung und zum Trainieren der räumlichen Desorientierung eingesetzt. Ein weiterer Einsatzbereich stellt das Testen des dynamischen Verhaltens von mechanischen Strukturen oder von Fluiden dar.

Am Deutschen Zentrum für Luft- und Raumfahrt (DLR) wird am Institut für Systemdynamik und Regelungstechnik eine neuartige 8-Achs Bewegungsplattform aufgebaut, die zukünftig für verschiedene Anwendungsbereiche eingesetzt werden soll. Hauptsächlich wird die Bewegungsplattform für die klassische Bewegungssimulation verwendet, zum Beispiel als Fahrsimulator oder Flugsimulator. Darüber hinaus sollen weitere Anwendungsbereiche wie zum Beispiel Schwapptests von Flugzeugtanks oder die Analyse des dynamischen Verhaltens von Satelliten untersucht werden.

Als Basis für die Bewegungsplattform dient ein Hexapod, auf dem zusätzlich zwei rotatorische Achsen und eine Gondel montiert werden. Die erste der beiden Achsen (Achse sieben) besteht aus einem Drehteller, welcher die Gondel um die Gierachse rotieren lässt. Mithilfe der zweiten Achse (Achse acht), welche um 90° gedreht zur Ersten angebracht ist, kann die Rollachse der Gondel bewegt werden. Durch die beiden zusätzlichen Achsen wird der rotatorische Arbeitsraum der Bewegungsplattform vergrößert, was die Einsatzmöglichkeiten des Systems erhöht.

Für die vorliegende Masterarbeit wurden zwei Ziele definiert. Das erste Ziel ist die Entwicklung eines Gelenkbahnplanungsmoduls, mit dem es möglich sein soll, unter Berücksichtigung von Positions-, Geschwindigkeits- und Beschleunigungslimits der Gelenke eine kartesische Trajektorie auf der Bewegungsplattform abzufahren. Um das Gelenkbahnplanungsmodul zu verifizieren, wird außerdem ein Mehrkörpersimulationsmodell und eine 3D-Visualisierung erstellt. Als zweites Ziel sollen die Antriebe für die beiden zusätzlichen Achsen ausgelegt werden.

Das Gelenkbahnplanungsmodul wird aufbauend auf der inversen Kinematik der 8-Achs Bewegungsplattform erstellt. Da es sich um ein redundantes System handelt, muss die inverse Kinematik über ein Optimierungsproblem auf Geschwindigkeitsebene berechnet werden [2]. Es besteht die Möglichkeit die inverse Kinematik so zu erweitern, dass mehrere Aufgaben mit verschiedenen Prioritäten definiert werden können [3, 50]. Darauf aufbauend wurde in einer Veröffentlichung von Masone u. a. [34] ein Gelenkbahnplanungsmodul für einen Bewegungssimulator erstellt, der auf einem seriellen Industrieroboter basiert. Ein Problem dabei ist, dass die Positionslimits nicht in die inverse Kinematik integriert sind, sondern erst auf deren Lösung angewendet werden. Folglich können die unbeschränkten Aktuatoren nicht optimal ausgenutzt werden. In dieser Arbeit wird ein Gelenkbahnplanungsmodul aufbauend auf der Veröffentlichung von Moe u. a. [35] erstellt. Damit können die Positionslimits direkt in die inverse Kinematik integriert werden und so die unbeschränkten Aktuatoren optimal zur Erfüllung der vorgegebenen Aufgaben eingesetzt werden.

Für die Antriebsauslegung der beiden zusätzlichen Achsen, wird mithilfe des Mehrkörpersimulationsmodells versucht, Lastfälle zu identifizieren, bei denen die Antriebsmomente maximal werden.

Aufbau der Arbeit

Zu Beginn werden in Kapitel 2 existierende Bewegungsplattformen und deren Steueralgorithmen vorgestellt, sowie der Aufbau der 8-Achs Bewegungsplattform erklärt und die Vor- und Nachteile davon diskutiert. Anschließend wird in Kapitel 3 auf die kinematische Modellbildung eingegangen. Darauf aufbauend wird in Kapitel 4 die Funktionsweise der Gelenkbahnplanung erläutert und in Kapitel 5 die Implementierung des Mehrkörpersimulationsmodells und der Gelenkbahnplanung beschrieben. Weiter wird in Kapitel 6 die Gelenkbahnplanung mithilfe des Mehrkörpersimulationsmodells verifiziert. Es folgt in Kapitel 7 die Antriebsauslegung von Achse 7 und 8. Kapitel 8 fasst schließlich die wichtigsten Aspekte der Arbeit zusammen und gibt einen Ausblick.

2. Grundlagen zu Bewegungsplattformen

In diesem Kapitel wird der Stand der Technik von Bewegungsplattformen dargestellt, die in der Bewegungssimulation zum Einsatz kommen. Zu Beginn werden in Abschnitt 2.1 verschiedene Strukturen für Bewegungsplattformen vorgestellt. In Abschnitt 2.2 folgt eine Erklärung der dafür verwendeten Steueralgorithmen. Schließlich wird in Abschnitt 2.3 die neuartige 8-Achs Bewegungsplattform präsentiert, welche in dieser Arbeit modelliert wird. Hierbei werden insbesondere die Vor- und Nachteile gegenüber dem Stand der Technik diskutiert.

2.1. Strukturen von Bewegungsplattformen

Um die nötigen Beschleunigungen, Geschwindigkeiten und Positionen für die oben genannten Anwendungen zu erreichen, muss die Bewegungsplattform eine geeignete Struktur aufweisen. Die Art der Struktur eines mechanischen Systems kann durch die Analyse der kinematischen Kette festgestellt werden. In Abbildung 1 sind drei elementare Roboterstrukturen für den planaren Fall abgebildet.

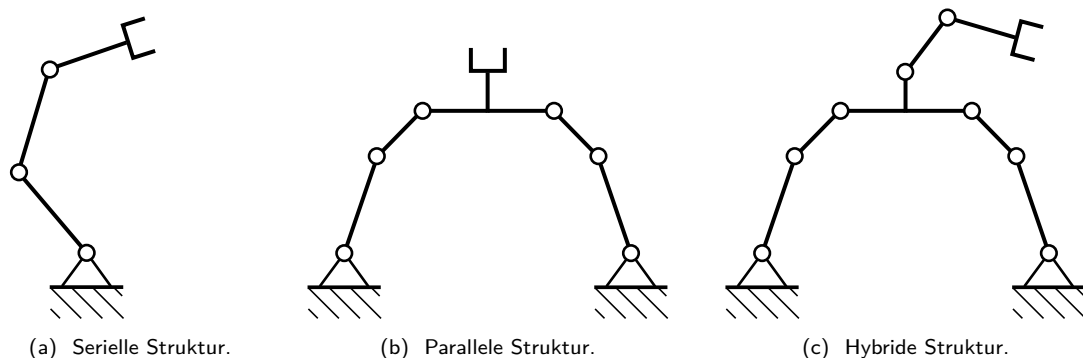


Abbildung 1 Elementare Roboterstrukturen

Die Struktur eines mechanischen Systems wird auch oft als Kinematik bezeichnet. In dieser Arbeit wird jedoch die Bezeichnung Struktur verwendet, um Verwechslungen mit der später definierten direkten und inversen Kinematik zu vermeiden. Bei einem System mit einer offenen kinematischen Kette, wie zum Beispiel bei einem klassischen Knickarmroboter, handelt es sich um eine serielle Struktur (Abb. 1a), oder serielle Kinematik. Liegt mindestens eine geschlossene kinematische Kette vor, besitzt das System eine parallele Struktur (Abb. 1b), welche auch oft als Parallelkinematik bezeichnet wird. Ein Beispiel dafür ist der Hexapod. Eine Struktur, die aus einer Kombination von seriellen und parallelen Strukturen besteht, wird als hybride Struktur (Abb. 1c) bezeichnet.

In vielen Simulatoren findet ein Hexapod (siehe Abbildung 2) Anwendung. Manchmal in Kombi-

nation mit zusätzlichen translatorischen oder rotatorischen Achsen. Ein Hexapod ist eine parallele Roboterstruktur mit sechs Freiheitsgraden, welche aus zwei Plattformen besteht, die über sechs Linearaktuatoren miteinander verbunden sind. Durch die parallele Bauweise können höhere Nutzlasten bewegt werden als mit einer seriellen Struktur, die zum Beispiel in Industrierobotern Anwendung findet. Weiter zeigt die parallele Bauweise auch eine höhere Steifigkeit und ein reduziertes Schwingungsverhalten. Es ist jedoch festzuhalten, dass im Gegensatz zu einer seriellen Struktur ein kleinerer Arbeitsraum zu Verfügung steht und ein erhöhter Platzbedarf nötig ist. Durch eine Bauweise mit geringem Gewicht können mit einem Hexapod auch gute dynamische Eigenschaften erreicht werden.



Abbildung 2 Beispiel für einen Hexapod, aus [9].

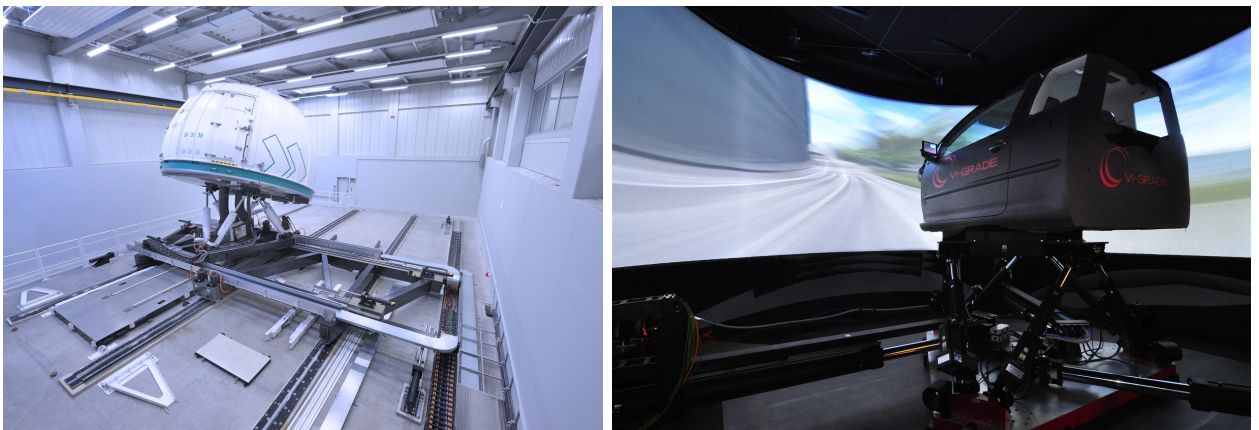
Die ersten Arbeiten zu Hexapoden wurden 1962 von Gough [24] und 1965 von Stewart [54] veröffentlicht, weshalb der Hexapod auch als Stewart-Plattform oder Gough-Stewart-Plattform bezeichnet wird. Zur historischen Entwicklung von parallelen Robotern sei auf Bonev [8] verwiesen.

In Fahrsimulatoren [30, 55] und in Flugsimulatoren, welche zum Beispiel für die Pilotenausbildung (Full-Flight-Simulatoren) oder in der Forschung [17] eingesetzt werden, wird in vielen Fällen ein Hexapod als Bewegungsplattform verwendet. Auf diesem Hexapod wird dann zum Beispiel ein Flugzeugcockpit oder ein Auto montiert und über ein bildgebendes und akustisches System eine realistische Umgebung geschaffen. Diese Simulatoren haben den Vorteil, dass es sich um nicht redundante Systeme handelt. Das bedeutet, dass der Freiheitsgrad des Simulators nicht größer als der Freiheitsgrad des Arbeitsraums ist. Dadurch kann das System im Vergleich zu redundanten Systemen einfacher gesteuert werden. Ein Nachteil dieser Bauweise ist der kleine Arbeitsraum, weshalb es nicht möglich ist, lang anhaltende Beschleunigungen zu erzeugen, die zum Beispiel beim Anfahren eines Autos vorkommen. Durch eine Täuschung des menschlichen Gleichgewichtssinns kann dieses Problem jedoch verringert werden.

Das menschliche Gleichgewichtsorgan besitzt die Fähigkeit Beschleunigungen wahrzunehmen, jedoch kann es nicht zwischen translatorischen Beschleunigungen und der Erdbeschleunigung unterscheiden. Es ist deshalb möglich, dem Gleichgewichtsorgan durch eine Neigung um eine horizontale Achse eine horizontale Beschleunigung vorzutäuschen. Das bedeutet, wenn eine Person in einem Bewegungssimulator sitzt und dieser nach hinten gekippt wird, dann hat diese Person das Gefühl nach vorne zu

beschleunigen, da sie durch die Erdbeschleunigung in den Sitz gedrückt wird. Dieser Vorgang wird in der Literatur als Tilt Coordination bezeichnet [18].

Um horizontale Beschleunigungen besser nachbilden zu können, kann der zuvor beschriebene Hexapod-Simulator auf einer oder auf zwei Linearachsen befestigt werden. Diese Bauweise wird vor allem in Fahrsimulatoren verwendet [1, 11, 47]. In Abbildung 3a ist ein Bewegungssimulator mit zwei Linearachsen dargestellt. Dabei handelt es sich um ein redundantes System mit insgesamt acht Freiheitsgraden. Durch die zusätzlichen Linearachsen können auch horizontale Beschleunigungen besser simuliert werden, welche durch eine Tilt Coordination nicht möglich sind, da diese nur begrenzt einsetzbar ist. Die begrenzte Einsetzbarkeit kommt daher, dass die Erdbeschleunigung einen konstanten Wert besitzt und durch Neigen des Simulators nur ein Teil davon genutzt wird. Außerdem führen zu große Neigungen zu einem Wahrnehmungsfehler, weil dadurch eine Drehbewegung entsteht, die es eigentlich nicht gibt.



(a) Beispiel für einen Fahrsimulator mit acht Freiheitsgraden, aus [47].

(b) Beispiel für einen Fahrsimulator mit neun Freiheitsgraden, aus [25].

Abbildung 3 Beispiele für Bewegungssimulatoren.

Weiter besteht die Möglichkeit, einen neunten Freiheitsgrad in Form eines Drehtellers auf dem Hexapod hinzuzufügen [33, 37]. Dieser führt zum Beispiel in Fahrsimulatoren zu einer verbesserten Nachbildung von Kurvenfahrten [41] und die sogenannte Simulatorkrankheit (engl. Simulator Sickness) kann verringert werden [42]. Bei der Simulatorkrankheit handelt es sich um ein Gefühl der Übelkeit, ähnlich wie bei der Seekrankheit. Diese tritt in Bewegungssimulatoren dann auf, wenn die Visualisierung nicht mit den vermittelten Bewegungseindrücken übereinstimmt.

Eine weitere Struktur für eine Bewegungsplattform mit neun Freiheitsgraden wird in Abbildung 3b dargestellt. Dort wird ein Hexapod auf einem planaren Dreibeingestell montiert. Das Dreibeingestell ist eine Plattform, welche über drei Luftpolster auf einer glatten Oberfläche am Boden gleitet und mithilfe von drei Linearaktuatoren bewegt werden kann. So kann es sich mit drei Freiheitsgraden in der Ebene bewegen und erfüllt die gleichen Funktionen wie zwei Linearachsen und ein Drehteller zusammen. Durch die parallele Struktur bietet das Dreibeingestell gute dynamische Eigenschaften. Da

die Länge der Linearaktuatoren nicht beliebig gewählt werden kann, gibt es eine weitere Bauweise, bei der als Basis anstelle des Dreibeingestells ein seilgetriebenes System verwendet wird. Dadurch kann ein größerer translatorischer Arbeitsraum realisiert werden.

Alle bis jetzt vorgestellten Bewegungssimulatoren bauen auf einem Hexapod auf. Es existieren aber auch Bewegungsplattformen, welche auf großen seriellen Industrierobotern basieren [6, 34]. Diese weisen jedoch im Gegensatz zu parallelen Systemen eine geringere Struktursteifigkeit, niedrigere Tragzahlen und eine schlechtere Dynamik auf. Dem Gegenüber bieten Industrieroboter aber eine höhere Bewegungsfreiheit, einen größeren Arbeitsraum und eine beliebige Positionierung des Endeffektors im Arbeitsraum. Dadurch ergeben sich interessante neue Möglichkeiten für die Bewegungssimulation, wie zum Beispiel Schleudermanöver oder Überkopfflüge. Auch hier gibt es die Möglichkeit den Roboter auf einer Linearachse zu befestigen, um den Arbeitsraum zu erweitern [6] (siehe Abbildung 4a), oder der Pilotenkabine die Möglichkeit zu geben, sich zusätzlich 90° zu drehen [34]. Dadurch resultiert jeweils eine Bewegungsplattform mit sieben Freiheitsgraden.



(a) Beispiel für einen Bewegungssimulator basierend auf einem seriellen Industrieroboter, aus [6].



(b) Beispiel für einen Bewegungssimulator basierend auf einer Zentrifuge, aus [21].

Abbildung 4 Beispiele für Bewegungssimulatoren.

Eine große Herausforderung stellt die Simulation von hochdynamischen Flugzeugen dar, da die auftretenden Beschleunigungen mit einem Hexapod oder einem Industrieroboter nicht erzeugt werden können. Eine Bewegungsplattform, welche es ermöglicht hohe Beschleunigungen nachzubilden, besteht aus einer Zentrifuge, auf der eine zweiachsig kardanisches gelagerte Kabine befestigt ist, die mithilfe von Motoren gedreht werden kann [16, 31] (siehe Abbildung 4b). Es ist dadurch möglich, Beschleunigungen in der Kabine zu erzeugen, die bis zu 15 Mal größer sind als die Erdbeschleunigung [16].

Weiter gibt es eine Bewegungsplattform, die aus einer drehbar gelagerten acht Meter langen horizontalen Linearachse besteht, auf der eine zwei Meter lange vertikale Linearachse montiert ist. Darauf ist wiederum eine dreiachsige kardanisches gelagerte Gondel befestigt. Dieses System besitzt insgesamt sechs Freiheitsgrade und wird zur Flugsimulation, Fahrsimulation und zum trainieren der

räumlichen Desorientierung eingesetzt [7].

Ein großes Problem von auf Zentrifugen basierten Bewegungssimulatoren ist die Bewegungskrankheit, welche die gleichen Symptome wie die Simulatorkrankheit aufweist. Diese wird durch eine Corioliskraft ausgelöst, die auftritt, wenn sich der Zentrifugenarm dreht, während gleichzeitig die kardanisch gelagerte Kabine gedreht wird oder der Pilot den Kopf bewegt [16].

2.2. Steueralgorithmen für Bewegungsplattformen

Neben der Struktur der Bewegungsplattform spielt auch der Steueralgorithmus eine wichtige Rolle. Soll die Bewegung eines Autos oder eines Flugzeugs mit einer Bewegungsplattform simuliert werden, dann können diese Bewegungen nicht eins zu eins übernommen werden, da der Arbeitsraum der Bewegungsplattform beschränkt ist. Der Prozess der Umwandlung einer gewünschten Bewegung in eine Bewegung, die von einer Bewegungsplattform ausgeführt werden kann, wird als Motion Cueing bezeichnet. Diese Umwandlung wird mithilfe von Motion Cueing Algorithmen (MCA) durchgeführt, die dafür sorgen, dass die dynamischen Limits der Bewegungsplattform nicht überschritten werden und der Arbeitsraum eingehalten wird. Das Ziel ist es, dass sich die vermittelten Bewegungseindrücke, welche durch einen MCA erzeugt werden, nicht von der realen Bewegung unterscheiden lassen. Der Grund, wieso sich die Bewegungen verändern lassen und trotzdem ein ähnliches Bewegungsgefühl entsteht, ist, dass der Mensch Bewegungen neben der auditiven und visuellen Wahrnehmung hauptsächlich als Beschleunigungen wahrnimmt. Zusätzlich werden diese Beschleunigungen erst ab einem gewissen Grenzwert wahrgenommen. Dadurch ist es möglich, zum Beispiel die weiter oben erwähnte Tilt Coordination anzuwenden oder gleichbleibende Geschwindigkeiten auszufiltern [18].

Grundlegend gibt es zwei verschiedene Arten von MCA. Das sind zum einen filterbasierte MCA und zum anderen MCA, welche auf der modellprädiktiven Regelung (engl. Model Predictive Control) aufbauen. Für eine ausführliche Übersicht existierender MCA sei auf die Arbeiten von Fischer [20], Ellensohn [18] und Garrett u. a. [22] verwiesen. In dieser Masterarbeit wird der Ansatz der filterbasierten MCA verwendet, weshalb im Folgenden genauer darauf eingegangen wird.

Zum allgemeinen Verständnis der verschiedenen filterbasierten MCA wird hier der klassische Washout Filter [18] beschrieben. Darin werden die vorgegebenen Beschleunigungssignale in hoch- und niederfrequente Signale aufgeteilt. Die hochfrequenten Signale werden über einen Washout Filter, welcher dafür sorgt, dass die Bewegungsplattform kontinuierlich in die Ausgangslage zurückgeführt wird, gefiltert. Die niederfrequenten translatorischen Beschleunigungen werden mithilfe der Tilt Coordination weiterverarbeitet. Da dieser Algorithmus keine Stellgrößenbeschränkungen enthält, müssen die Signale durch Einstellen der Parameter beschränkt werden.

Klassische, filterbasierte MCA können für nicht redundante Bewegungsplattformen wie den Hexapod sehr einfach zur Steuerung der Motoren eingesetzt werden. Handelt es sich jedoch um ein redundantes System, wird es komplizierter, da das System mehr Freiheitsgrade als der Arbeitsraum besitzt. Eine Lösung besteht darin, einen Bandpassfilter zu integrieren, der einen gewissen Beschleunigungsbereich extrahiert und diesen einem einzelnen Aktuator zuweist [18]. Betrachtet man das Beispiel des Bewegungssimulators aus Abbildung 3a, dann kann ein Bandpassfilter einen Bereich der horizontalen Beschleunigungen herausfiltern und daraus direkt Steuersignale für die beiden Linearachsen

erzeugen. Die hohen und niedrigen Beschleunigungen können wiederum an den Hexapod übergeben werden.

Eine weitere Lösung stellt eine optimierungsbasierte inverse Kinematik dar, welcher die Stellsignale aus dem Washout Filter übergeben werden. Diese wird hauptsächlich für Bewegungssimulatoren eingesetzt, die auf seriellen Industrierobotern basieren [6, 23]. Aufgrund der seriellen Kinematik spielt dabei die Vermeidung von Singularitäten eine große Rolle und es ist damit möglich, Stellgrößenbeschränkungen einzuführen.

2.3. 8-Achsen Bewegungsplattform

In diesem Abschnitt wird die neuartige 8-Achs Bewegungsplattform vorgestellt. Zu Beginn wird der Aufbau des Systems beschrieben und im Anschluss wird das System mit dem Stand der Technik verglichen.

Da das System noch nicht fertig ausgelegt ist, kann es noch zu Veränderungen in der Bauweise kommen. An der kinematischen Struktur wird sich jedoch nichts mehr ändern.

2.3.1. Systembeschreibung

In Abbildung 5 ist der Entwurf der 8-Achs Bewegungsplattform in Vorder- und Seitenansicht dargestellt. Als Basis für die Plattform dient ein Hexapod (siehe Abbildung 2), welcher um zwei rotatorische Achsen (Achse 7 und Achse 8) erweitert wird. Auf Achse 8 ist eine Gondel montiert, die für bis zu zwei Personen Platz bietet. Der auf Achse 7 befestigte Drehteller ermöglicht eine Drehung der Gondel um die Hochachse. Durch die beiden um 90 Grad zueinander angeordneten Achsen vergrößert sich der rotatorische Arbeitsraum der Bewegungsplattform. Außerdem wird beiden Achsen durch einen Schleifring eine endlose Drehung ermöglicht.

Für die gesamte Bewegungsplattform ergeben sich aus den sechs Freiheitsgraden des Hexapods und den beiden zusätzlichen Achsen insgesamt acht Freiheitsgrade. Es handelt sich folglich um ein redundantes System. Diese Redundanz stellt für die kinematische Modellbildung in Kapitel 3 eine besondere Herausforderung dar.

Des Weiteren ist festzuhalten, dass die Bewegungsplattform so konstruiert ist, dass sich der Aufbau auf dem Drehteller verändern lässt. Es besteht die Möglichkeit, die Gondel sowie die Struktur auf der Achse 8 befestigt ist (siehe Abbildung 5), von der Plattform zu entfernen und durch andere Komponenten zu ersetzen. Ziel dabei ist das System in Zukunft nicht nur als Bewegungssimulator zu verwenden, sondern auch für weitere Einsatzbereiche nutzen zu können. Zum Beispiel um Schwapp-tests (engl. sloshing test) für Flugzeugtanks durchzuführen oder um das dynamische Verhalten von mechanischen Strukturen zu testen.

2.3.2. Vor- und Nachteile der Bewegungsplattform

In diesem Abschnitt werden die Vor- und Nachteile der 8-Achs Bewegungsplattform gegenüber vergleichbarer Systemen aus Abschnitt 2.1 diskutiert. Aufgrund großer Ähnlichkeit in Arbeitsraum und Aufbau wird dafür die Bewegungsplattform auf Basis eines seriellen Industrieroboters (siehe Abb. 4a) und die Bewegungsplattform auf Basis eines planaren Dreibeingestells in Kombination mit einem

Hexapod (siehe Abb. 3b) verwendet.

Im Folgenden werden die Vor- und Nachteile der 8-Achs Bewegungsplattform gegenüber des Systems auf Basis eines seriellen Industrieroboters erklärt.

Durch den Hexapod besitzt die neue Bewegungsplattform eine höhere Steifigkeit und kann somit dynamischere Bewegungen erzeugen. Außerdem kann durch den Hexapod auch eine höhere Nutzlast bewegt werden, wodurch sich die Einsatzmöglichkeiten erweitern. Ein Beispiel dafür sind Schwapp-tests von Flugzeugtanks. Weiter besitzt die neue Plattform keine Singularitäten im Arbeitsraum (siehe Kapitel 3), wodurch die Bahnplanung vereinfacht wird. Es ist jedoch festzuhalten, dass die Berechnung der Kinematik aufgrund der parallelen Struktur des Hexapods rechenintensiver wird. Ein weiterer Nachteil ist die verringerte Bewegungsfreiheit bzw. der kleinere Arbeitsraum.

Weiter werden die Vor- und Nachteile der 8-Achs Bewegungsplattform gegenüber des Systems auf Basis eines planaren Dreibeingestells in Kombination mit einem Hexapod erläutert.

Die neue Bewegungsplattform zeichnet sich durch einen einfacheren Aufbau und einen geringeren Platzbedarf aus. Durch die beiden zusätzlichen Achsen besitzt diese außerdem einen größeren rotatorischen Arbeitsraum. Der translatorische Arbeitsraum ist jedoch kleiner. Es soll aber festgehalten werden, dass durch die Bauweise der neuen Bewegungsplattform auch der translatorische Arbeitsraum in Richtung der Querachse vergrößert wird. Kippt der Hexapod auf die Seite, kann durch eine Drehung von Achse 8 die Gondel immer horizontal ausgerichtet werden, wodurch sich eine translatorische Bewegung in der Querachse ergibt.

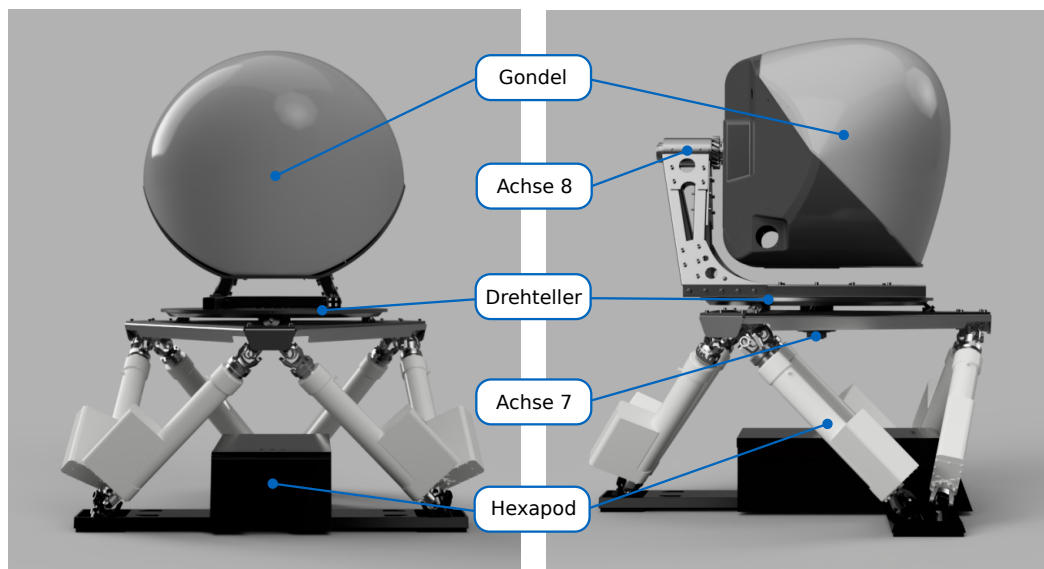


Abbildung 5 Aufbau der 8-Achs Bewegungsplattform.

3. Kinematische Modellbildung

Dieses Kapitel beschäftigt sich mit der kinematischen Modellierung der 8-Achs Bewegungsplattform. Dafür werden in Abschnitt 3.1 wichtige Grundlagen der Kinematik erklärt. Darauf folgend wird zunächst in Abschnitt 3.2 die Kinematik für den seriellen Teil der Bewegungsplattform erstellt und danach in Abschnitt 3.3 für den parallelen Teil. Anschließend werden die beiden Teile in Abschnitt 3.4 miteinander verknüpft. Weiter gibt Abschnitt 3.5 eine Erklärung der inversen Kinematik, in der mehrere Aufgaben mit verschiedenen Prioritäten definiert werden können. Zuletzt wird in Abschnitt 3.6 die inverse Kinematik durch eine Rückführung der gemessenen Minimalkoordinaten \mathbf{q} erweitert.

3.1. Kinematische Grundlagen

In diesem Abschnitt wird zu Beginn die Struktur des Systems untersucht. Darauf folgt die Definition von Parametern und Koordinatensystemen, welche die Kinematik beschreiben. Im Anschluss werden auftretende Singularitäten genauer analysiert und die in dieser Arbeit verwendete Nomenklatur angegeben.

Struktur. In Kapitel 2 (Abb. 1) wurden zu Beginn bereits drei elementare Roboterstrukturen vorgestellt. Bei der 8-Achs Bewegungsplattform handelt es sich um eine hybride Struktur. Das ergibt sich aus der Kombination der parallelen Struktur des Hexapods und der seriellen Struktur der beiden zusätzlichen Achsen.

Parameter. Die Kinematik eines Systems kann auf unterschiedliche Arten beschrieben werden. Dafür werden typischerweise zwei verschiedene Möglichkeiten verwendet, die Darstellung im Konfigurationsraum oder im Arbeitsraum.

Im Konfigurationsraum wird die Kinematik mithilfe der Minimalkoordinaten $\mathbf{q} \in \mathbb{R}^n$ beschrieben. Die Dimension des Konfigurationsraums ist gleich der Anzahl der Freiheitsgrade des Systems und somit gleich der Anzahl der Aktuatoren. Minimalkoordinaten stellen einen minimalen Satz an Parametern dar, welche die Konfiguration eines mechanischen Systems eindeutig beschreiben. Üblicherweise werden dafür die Stellgrößen verwendet, also die Gelenkwinkel oder die Verschiebungen der Aktuatoren. Ein Vorteil der Minimalkoordinaten ist, dass diese meist direkt im mechanischen System gemessen werden können.

Im Arbeitsraum wird die Kinematik mithilfe der Arbeitsraumkoordinaten $\mathbf{w} \in \mathbb{R}^m$ beschrieben:

$$\mathbf{w} = \begin{pmatrix} \mathbf{r} \\ \boldsymbol{\beta} \end{pmatrix}. \quad (3.1)$$

Diese definieren die Pose, also die Position $\mathbf{r} \in \mathbb{R}^3$ und die Orientierung $\mathbf{A}(\boldsymbol{\beta}) \in \mathbb{R}^{3 \times 3}$ des Endeffektors (oder eines beliebigen Körpers) bezüglich eines Inertialsystems. Die Orientierung wird durch eine Rotationsmatrix $\mathbf{A}(\boldsymbol{\beta})$, welche mit den Rotationsparametern $\boldsymbol{\beta}$ parametrisiert ist, dargestellt. Die ein-

zelen Spalten der Rotationsmatrix stellen die Einheitsvektoren des beschriebenen Koordinatensystems dar. Rotationsparameter beschreiben eine eindeutige Orientierung des Koordinatensystems und können auf unterschiedliche Weise definiert werden. Beispielsweise durch eulersche Winkel, Kardan-Winkel oder Quaternionen. Für eine detaillierte Beschreibung der verschiedenen Rotationsparameter sei auf [52] verwiesen. In dieser Arbeit werden als Rotationsparameter Quaternionen verwendet, um Repräsentationssingularitäten (engl. representation singularity) zu vermeiden [53].

Der Vorteil der Darstellung im Arbeitsraum ist, dass sich dadurch Aufgaben einfacher beschreiben lassen. Ein Beispiel dafür ist die Vorgabe einer Trajektorie für den Endeffektor. Diese lässt sich durch die Position und Orientierung direkt vorgeben.

Quaternionen wurden von Sir William Hamilton beschrieben, weshalb die Menge der Quaternionen meist mit \mathbb{H} bezeichnet wird. In dieser Arbeit werden ausschließlich Einheitsquaternionen verwendet, welche wie folgt definiert sind:

$$\mathbf{Q} = \begin{pmatrix} \eta \\ \boldsymbol{\epsilon} \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ \mathbf{n} \sin\left(\frac{\theta}{2}\right) \end{pmatrix}, \quad (3.2)$$

mit der Bedingung $\eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 = 1$. Dabei beschreibt $\mathbf{n} \in \mathbb{R}^3$ einen Einheitsvektor, welcher die Rotationsachse für die Rotation eines Koordinatensystems um den Winkel θ darstellt.

Um Rotationen zu kombinieren, können Quaternionen multipliziert werden:

$$\mathbf{Q}_1 * \mathbf{Q}_2 = \begin{pmatrix} \eta_1 \eta_2 - \boldsymbol{\epsilon}_1 \cdot \boldsymbol{\epsilon}_2 \\ \eta_1 \boldsymbol{\epsilon}_2 + \eta_2 \boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_1 \times \boldsymbol{\epsilon}_2 \end{pmatrix}. \quad (3.3)$$

Durch die Multiplikation wird eine Drehung \mathbf{Q}_2 gefolgt von der Drehung \mathbf{Q}_1 durchgeführt. Die Inverse eines Quaternion ist durch $\mathbf{Q}^{-1} = (\eta, -\boldsymbol{\epsilon})$ gegeben. Für weitere Eigenschaften und Umrechnungen von Quaternionen sei auf [52] und [26] verwiesen.

Direkte und inverse Kinematik. Die sogenannte direkte und inverse Kinematik ermöglicht eine Transformation zwischen dem Konfigurationsraum und dem Arbeitsraum. Mithilfe der direkten Kinematik kann die Pose des Endeffektors (oder eines beliebigen Körpers) aus den Minimalkoordinaten berechnet werden. Die inverse Kinematik stellt das Gegenstück zur direkten Kinematik dar. Das bedeutet, dass damit die Minimalkoordinaten aus den Arbeitsraumkoordinaten berechnet werden können. Kurz zusammengefasst:

- Direkte Kinematik: $\mathbf{w} = f(\mathbf{q})$,
- Inverse Kinematik: $\mathbf{q} = f^{-1}(\mathbf{w})$.

Koordinatensysteme. Zur Berechnung der Kinematik der 8-Achs Bewegungsplattform wird ein Inertialsystem eingeführt sowie für jeden Körper ein körperfestes Koordinatensystem. Diese sind in Abbildung 6 dargestellt. Bei dem Inertialsystem handelt es sich in der Robotik üblicherweise um ein Koordinatensystem, welches fest mit der Basis des Roboters verbunden ist.

Im Folgenden wird kurz die Namensgebung und die Ausrichtung der Koordinatensysteme erklärt. Das Inertialsystem wird mit dem Kürzel B für „base platform“ bezeichnet. Das P -Koordinatensystem

beschreibt die „payload platform“, also die obere bewegte Plattform des Hexapods. Für den Drehteller (engl. rotary table) wurde das R -Koordinatensystem definiert und für die Pilotenkabine (engl. cabin) das C -Koordinatensystem. Mithilfe des T -Koordinatensystems wird schließlich der Endeffektor (engl. Tool Center Point, kurz TCP) beschrieben.

Im Weiteren wird näher auf die Ausrichtung der Koordinatensysteme eingegangen. Die Koordinatensysteme B und P sind im Steuercomputer des Hexapods festgelegt. Deshalb wurden diese an die gleiche Stelle gesetzt, wie in der Steuerung definiert. Das Koordinatensystem für den Endeffektor wurde so ausgerichtet, wie in der Fahrsimulation üblich. Das bedeutet, dass die z -Achse nach oben und die x -Achse nach vorne zeigt. Weiter wurden die Koordinatensysteme R und C so orientiert, dass diese mithilfe von Denavit-Hartenberg Rotationsparametern beschrieben werden können. Für das Mehrkörpersimulationsmodell werden die Koordinatensysteme wie in Abbildung 6 dargestellt verwendet. Belastungen können dadurch direkt an den Achsflanschen ermittelt werden. Für die Berechnung der Kinematik, welche in der Bahnplanung verwendet wird, wird das Koordinatensystem R in der z -Achse verschoben, um eine vollständige Beschreibung mit Denavit-Hartenberg Parametern zu ermöglichen. Dies wird in Abschnitt 3.2 genauer erklärt.

Die relativen Verschiebungen zwischen den Koordinatensystemen werden mit den Variablen l und s beschrieben.

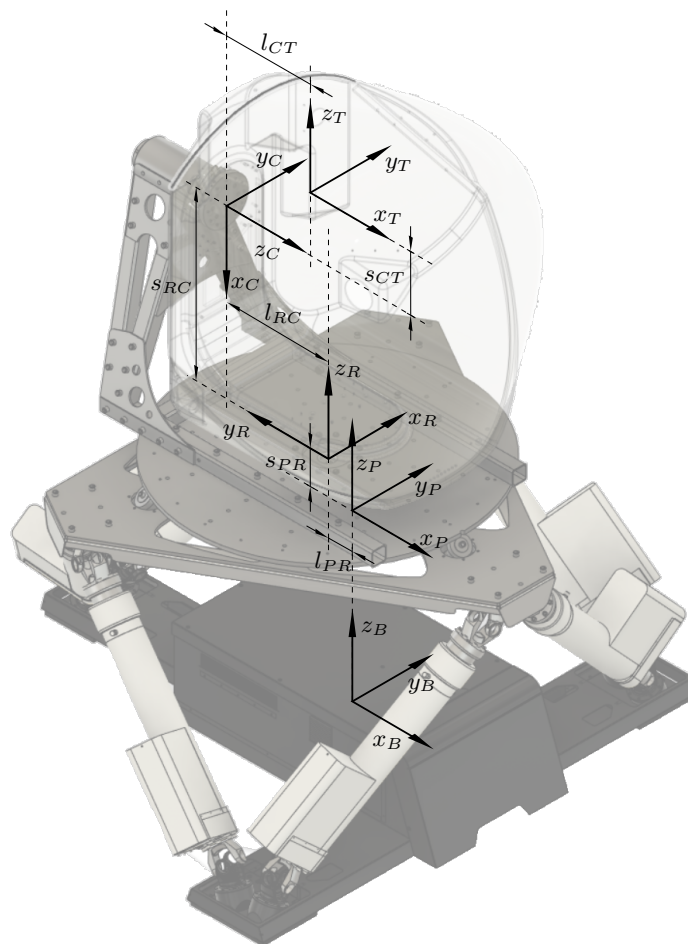


Abbildung 6 Koordinatensysteme der 8-Achs Bewegungsplattform.

Singularitäten. Ein weiterer wichtiger Punkt, der bei kinematischen Strukturen analysiert werden muss, sind Singularitäten. Diese treten bei Konfigurationen eines Systems auf, in denen die Jacobi-Matrix keinen vollen Rang besitzt [52].

Singularitäten können in drei verschiedene Gruppen eingeteilt werden. Die erste davon sind Randsingularitäten (engl. boundary singularity), welche bei der Erreichung von Aktuatorgrenzen auftreten. Diese können relativ einfach durch eine Aktuatorbeschränkung verhindert werden. Weiter können Arbeitsraumsingularitäten (engl. internal singularity) auftreten, wenn zum Beispiel zwei Rotationsachsen deckungsgleich ausgerichtet sind. Diese stellen die problematischste Gruppe der Singularitäten dar, da sie mitten im Arbeitsraum auftreten können. Es gibt außerdem Repräsentationssingularitäten, die durch Umrechnung der Orientierung $\mathbf{A}(\beta)$ in Rotationsparameter wie Kardan-Winkel oder eulersche Winkel auftreten (Gimbal Lock). Diese lassen sich durch eine Verwendung von Quaternionen als Rotationsparameter ganz einfach vermeiden. [52]

In [18] werden die Singularitäten von Hexapods analysiert. Diese können Randsingularitäten und Arbeitsraumsingularitäten besitzen. Arbeitsraumsingularitäten treten dann auf, wenn die Achse eines Aktuators parallel zur oberen Plattform des Hexapods ist. Zur Vermeidung von Randsingularitäten werden in dieser Arbeit Aktuatorbeschränkungen genutzt. Durch diese Beschränkungen und der Bauweise des Hexapods, welcher in der Bewegungsplattform verwendet wird, ist es auch nicht möglich, dass Arbeitsraumsingularitäten auftreten. Weiter besitzt auch die serielle Struktur keine singuläre Konfiguration, da die zwei Rotationsachsen immer 90-Grad zueinander ausgerichtet sind. Aus diesen Gründen wird für die Bewegungsplattform ohne Beweis davon ausgegangen, dass keine Singularitäten auftreten.

In Tabelle 1 ist die Nomenklatur für die kinematischen Größen angegeben, welche in dieser Arbeit verwendet wird. Der linke Index beschreibt in allen Variablen das Koordinatensystem, in dem diese definiert sind. Die Position eines Körpers ist durch den Ursprung des zugehörigen körperfesten Koordinatensystems definiert.

Tabelle 1 Nomenklatur der kinematischen Größen

${}^j\mathbf{r}_i$	Absolute Position des i -ten Körpers bezüglich des Inertialsystems, dargestellt im j -ten Koordinatensystem.
${}^j\mathbf{r}_{ik}$	Relative Position von Körper k zu Körper i , dargestellt im j -ten Koordinatensystem.
${}^j\mathbf{r}_{iP}$	Relative Position des Punktes P zu Körper i , dargestellt im j -ten Koordinatensystem.
${}^j\mathbf{A}_i$	Rotationsmatrix, welche eine Transformation vom i -ten in das j -te Koordinatensystem beschreibt.
${}^j\mathbf{J}_i$	Jacobi Matrix des i -ten Körpers, dargestellt im j -ten Koordinatensystem.

Für eine detaillierte Beschreibung der Kinematik von mechanischen Systemen sei auf das Buch von Siciliano u. a. [52] verwiesen.

3.2. Serielle Struktur

In diesem Abschnitt wird die Kinematik für den seriellen Teil der 8-Achs Bewegungsplattform berechnet. Dieser besteht aus einer Struktur mit zwei Drehgelenken und ist in Abbildung 7 schematisch dargestellt.

Die Beschreibung der direkten und inversen Kinematik wurde in Anlehnung an [52] und [12] erstellt.

3.2.1. Direkte Kinematik

Serielle Strukturen sind durch eine offene kinematische Kette gekennzeichnet. Jeder Körper besitzt dadurch immer nur genau einen Vorgänger. Aus diesem Grund bietet sich eine rekursive Berechnung der direkten Kinematik an. Dabei wird beginnend von der Basis des Roboters von Körper zu Körper gerechnet, bis der Endeffektor erreicht ist.

Um die rekursive Vorgehensweise besser beschreiben zu können, werden die Koordinatensysteme der Körper nummeriert. Begonnen wird im Inertialsystem mit 0 und anschließend von Körper zu Körper um eins hochgezählt.

Für die rekursive Berechnung der direkten Kinematik können folgende Gleichungen verwendet werden [46]:

$${}^i\mathbf{r}_i = {}^i\mathbf{A}_{v(i)} {}^{v(i)}\mathbf{r}_{v(i)} + {}^i\mathbf{r}_{v(i)i}, \quad (3.4)$$

$${}^0\mathbf{A}_i = {}^0\mathbf{A}_{v(i)} {}^{v(i)}\mathbf{A}_i. \quad (3.5)$$

In allen folgenden Gleichungen beschreibt der Index $v(i)$ den Vorgänger von Körper i , also Körper $i - 1$. In Gleichung (3.4) wird die Position ${}^i\mathbf{r}_i$ durch Transformation der Position ${}^{v(i)}\mathbf{r}_{v(i)}$ mit der Rotationsmatrix ${}^i\mathbf{A}_{v(i)}$ und anschließender Addition der Relativposition ${}^i\mathbf{r}_{v(i)i}$ berechnet. Auf diese Art können alle kinematischen Größen wie zum Beispiel Geschwindigkeit, Beschleunigung oder Winkelgeschwindigkeit rekursiv berechnet werden.

Um die Berechnung der direkten Kinematik zu vereinfachen, kann die sogenannte homogene Transformation verwendet werden [52]. Damit wird die relative Verschiebung und Verdrehung zwischen Körper i und Körper $v(i)$ kompakt in einer Matrix ${}^{v(i)}\mathbf{T}_i \in \mathbb{R}^{4 \times 4}$ dargestellt:

$$\begin{pmatrix} {}^{v(i)}\mathbf{r}_{v(i)P} \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} {}^{v(i)}\mathbf{A}_i & {}^{v(i)}\mathbf{r}_{v(i)i} \\ \mathbf{0}^T & 1 \end{pmatrix}}_{{}^{v(i)}\mathbf{T}_i} \begin{pmatrix} {}^i\mathbf{r}_{iP} \\ 1 \end{pmatrix}. \quad (3.6)$$

Durch Aneinanderreihung mehrerer homogener Transformationen

$${}^0\mathbf{T}_i = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \dots {}^{v(i)}\mathbf{T}_i = \begin{pmatrix} {}^0\mathbf{A}_i & {}^0\mathbf{r}_{0i} \\ \mathbf{0}^T & 1 \end{pmatrix} \quad (3.7)$$

kann die Transformation ${}^0\mathbf{T}_i$ zwischen beliebig vielen Körpern berechnet werden. Die gesamte Verschiebung ${}^0\mathbf{r}_{0i}$ und Verdrehung ${}^0\mathbf{A}_i$ zwischen den Körpern kann direkt aus (3.7) abgelesen werden. Es ist außerdem festzuhalten, dass mit $\mathbf{0} \in \mathbb{R}^3$ der Nullvektor beschrieben wird.

Eine standardisierte Methode zur Berechnung der direkten Kinematik mithilfe von homogenen Transformationen ist die Denavit-Hartenberg-Konvention (DH-Konvention). Dabei werden zur eindeutigen Beschreibung der relativen Verschiebung und Verdrehung von zwei Körpern nur vier Parameter benötigt. Normalerweise sind dafür sechs Parameter notwendig. Im Folgenden werden wichtige Aspekte der DH-Konvention kurz vorgestellt, um das Vorgehen in der Arbeit zu erklären. Für eine genauere Definition der Konvention sei auf [12] verwiesen. Die DH-Parameter lauten wie folgt:

- α_i : Verdrehung zwischen der z_{i-1} und z_i -Achse um die x_{i-1} -Achse.
- a_i : Verschiebung zwischen der z_{i-1} und z_i -Achse entlang der x_{i-1} -Achse.
- d_i : Verschiebung zwischen der x_{i-1} und x_i -Achse entlang der z_i -Achse.
- θ_i : Verdrehung zwischen der x_{i-1} und x_i -Achse um die z_i -Achse.

Werden die Verdrehungen und Verschiebungen der vier Parameter in der oben genannten Reihenfolge ausgeführt, erhält man eine Transformation von Körper $v(i)$ zu Körper i . Für die rekursive Berechnung der direkten Kinematik wird jedoch eine Transformation von Körper i zu Körper $v(i)$ benötigt. Diese ist im Folgenden angegeben¹[12]:

$$\begin{aligned} {}_{v(i)}\mathbf{T}_i &= \begin{pmatrix} \mathbf{A}_x^T(\alpha_i)\mathbf{A}_z^T(\theta_i) & \mathbf{A}_x^T(\alpha_i)(a_i\mathbf{e}_x + d_i\mathbf{e}_z) \\ \mathbf{0}^T & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_i \\ \sin(\theta_i)\cos(\alpha_i) & \cos(\theta_i)\cos(\alpha_i) & -\sin(\alpha_i) & -\sin(\alpha_i)d_i \\ \sin(\theta_i)\sin(\alpha_i) & \cos(\theta_i)\sin(\alpha_i) & \cos(\alpha_i) & \cos(\alpha_i)d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}. \end{aligned} \quad (3.8)$$

In Abbildung 7 sind die Koordinatensysteme für den seriellen Teil der 8-Achs Bewegungsplattform dargestellt, die zur Berechnung der direkten Kinematik mit DH-Parametern verwendet werden. Beim Koordinatensystem I handelt es sich um das Inertialsystem. Dieses wird später, bei der Zusammenführung der Lösungen für den seriellen und den parallelen Teil der 8-Achs Bewegungsplattform, durch das Koordinatensystem P des Hexapods ersetzt. Um die DH-Parameter verwenden zu können, muss das Koordinatensystem R , welches in Abbildung 7 in grau eingezeichnet ist, entlang der z-Achse verschoben werden. Dieses Koordinatensystem wird als R' bezeichnet. Obwohl dadurch die Koordinatensysteme mittels DH-Parameter darstellbar sind, wurde die DH-Konvention nicht vollständig eingehalten. Dies ist dadurch begründet, dass eine Orientierung für das Koordinatensystem von Achse 8 und des Inertialsystems in dieser Masterarbeit vorgegeben wurde. Normalerweise werden diese nicht vorgegeben, sondern ergeben sich erst durch die DH-Konvention. An dieser Stelle sei erwähnt, dass es durchaus vorkommen kann, dass für eine andere serielle Struktur mit festgelegter Anfangs- und Endorientierung keine Beschreibung durch DH-Parameter möglich ist.

Die DH-Parameter für den seriellen Teil der 8-Achs Bewegungsplattform sind in Tabelle 2 angegeben.

¹ Die Rotationsmatrizen $\mathbf{A}_x, \mathbf{A}_y$ und \mathbf{A}_z stellen Elementardrehungen um die einzelnen kartesischen Koordinatenachsen dar und die Vektoren $\mathbf{e}_x, \mathbf{e}_y$ und \mathbf{e}_z Einheitsvektoren, welche in Richtung der kartesischen Koordinatenachsen zeigen. Wie diese definiert sind, ist in Anhang A dargestellt.

Tabelle 2 Denavit-Hartenberg Parameter des seriellen Teils der 8-Achs Bewegungsplattform.

i	α_i	a_i	d_i	θ_i
R'	0	$-l_{PR}$	$s_{PR} + s_{RC}$	$\frac{\pi}{2} + \theta_R$
C	$\frac{\pi}{2}$	0	$-l_{RC}$	$-\frac{\pi}{2} + \theta_C$

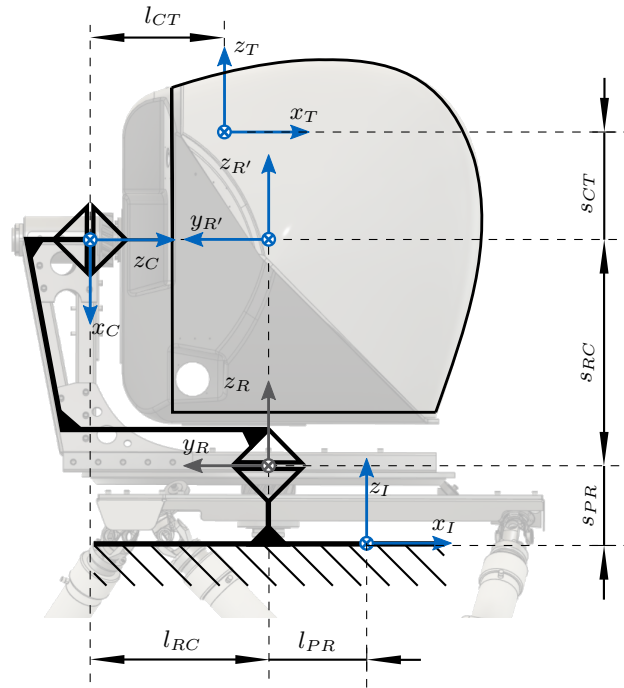


Abbildung 7 Schema des seriellen Anteils der 8-Achs Bewegungsplattform mit Koordinatensystemen und kinematischen Abmessungen.

Werden die DH-Parameter in die homogene Transformation aus (3.8) eingesetzt und wie in (3.7) multipliziert, so ergibt sich folgende homogene Transformation

$${}^I\mathbf{T}_C(\mathbf{q}_s) = \begin{pmatrix} {}^I\mathbf{A}_C(\mathbf{q}_s) & {}^I\mathbf{r}_{IC}(\mathbf{q}_s) \\ \mathbf{0}^T & 1 \end{pmatrix}, \quad (3.9)$$

die von den Minimalkoordinaten der seriellen Struktur $\mathbf{q}_s = (\theta_R, \theta_C) \in \mathbb{R}^2$ abhängt.

Durch Hinzufügen der homogenen Transformation ${}^C\mathbf{T}_T$ zu (3.9) resultiert die gesamte Verschiebung und Verdrehung zwischen dem Inertialsystem und dem Endeffektor:

$${}^I\mathbf{T}_T(\mathbf{q}_s) = {}^I\mathbf{T}_C(\mathbf{q}_s){}^C\mathbf{T}_T = \begin{pmatrix} {}^I\mathbf{A}_T(\mathbf{q}_s) & {}^I\mathbf{r}_{IT}(\mathbf{q}_s) \\ \mathbf{0}^T & 1 \end{pmatrix}. \quad (3.10)$$

Die homogene Transformation ${}^C\mathbf{T}_T$ ist folgendermaßen definiert:

$${}^C\mathbf{T}_T = \begin{pmatrix} \mathbf{A}_y(\frac{\pi}{2}) & {}^C\mathbf{r}_{CT} \\ \mathbf{0}^T & 1 \end{pmatrix}. \quad (3.11)$$

3.2.2. Inverse Kinematik

Zur Lösung der inversen Kinematik gibt es mehrere Möglichkeiten. Da es sich beim seriellen Teil des zu simulierenden Systems um eine relativ einfache Struktur handelt, könnte durch geometrische Überlegungen eine analytische Lösung für die inverse Kinematik gefunden werden. Diese Lösung ist jedoch nicht zielführend, da der serielle Teil später mit dem parallelen Teil der 8-Achs Bewegungsplattform verbunden werden soll. Die inverse Kinematik muss auf Positionsebene immer für das Gesamtsystem berechnet werden und kann nicht aus mehreren Lösungen für einzelne Teilstrukturen eines Systems zusammengesetzt werden.

Die Berechnung der inversen Kinematik kann außerdem auf Geschwindigkeitsebene erfolgen, welche auch als differenzielle inverse Kinematik bezeichnet wird. Dabei vereinfachen sich die kinematischen Beziehungen auf lineare Zusammenhänge. Weiter ist es auf Geschwindigkeitsebene im Gegensatz zur Positionsebene möglich, mehrere Teillösungen der inversen Kinematik mithilfe der Jacobi-Matrix miteinander zu verbinden. Die differenzielle Kinematik beschreibt den Zusammenhang zwischen der translatorischen Geschwindigkeit $\dot{\mathbf{r}} \in \mathbb{R}^3$ und der Winkelgeschwindigkeit $\boldsymbol{\omega} \in \mathbb{R}^3$, welche im Geschwindigkeitsvektor $\dot{\mathbf{x}} \in \mathbb{R}^6$ zusammengefasst werden, mit der Gelenkgeschwindigkeit $\dot{\mathbf{q}} \in \mathbb{R}^n$. Der Zusammenhang ist durch

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\mathbf{r}} \\ \boldsymbol{\omega} \end{pmatrix} = \mathbf{J}_g \dot{\mathbf{q}} \quad (3.12)$$

gegeben, worin $\mathbf{J}_g \in \mathbb{R}^{6 \times n}$ die geometrische Jacobi-Matrix darstellt [52]. Diese setzt sich aus der Jacobi-Matrix der Translation $\mathbf{J}_T \in \mathbb{R}^{3 \times n}$ und der Jacobi-Matrix der Rotation $\mathbf{J}_{R,\omega} \in \mathbb{R}^{3 \times n}$ bezogen auf die Winkelgeschwindigkeit $\boldsymbol{\omega}$ zusammen:

$$\mathbf{J}_g = \begin{pmatrix} \mathbf{J}_T \\ \mathbf{J}_{R,\omega} \end{pmatrix}. \quad (3.13)$$

Die Berechnung der Gelenkgeschwindigkeiten $\dot{\mathbf{q}}$ kann durch Inversion der Jacobi-Matrix berechnet werden. Handelt es sich bei dem System um eine redundante Struktur, wie die 8-Achs Bewegungsplattform, kann keine Inverse gebildet werden, sondern es muss auf andere Methoden zurückgegriffen werden. Dies wird in Abschnitt 3.4.2 genauer beschrieben.

Die Jacobi-Matrix kann wie alle anderen kinematischen Größen für serielle Strukturen rekursiv bestimmt werden. Die Jacobi-Matrix der Translation ${}^i\mathbf{J}_{T,i}$ und die Jacobi-Matrix der Rotation ${}^i\mathbf{J}_{R,\omega,i}$ ergeben sich zu

$${}^i\mathbf{J}_{T,i} = \frac{\partial {}^i\mathbf{v}_i}{\partial \dot{\mathbf{q}}} = {}^i\mathbf{A}_{v(i)} {}^v(i)\mathbf{J}_{T,v(i)} + \underbrace{\frac{\partial {}^i\mathbf{v}_{v(i)i}}{\partial \dot{\mathbf{q}}}}_{:= {}^i\mathbf{J}_{T,rel,i}}, \quad (3.14)$$

$${}^i\mathbf{J}_{R,\omega,i} = \frac{\partial {}^i\boldsymbol{\omega}_i}{\partial \dot{\mathbf{q}}} = {}^i\mathbf{A}_{v(i)} {}^v(i)\mathbf{J}_{R,v(i)} + \underbrace{\frac{\partial {}^i\boldsymbol{\omega}_{v(i)i}}{\partial \dot{\mathbf{q}}}}_{:= {}^i\mathbf{J}_{R,\omega,rel,i}}. \quad (3.15)$$

Der relative Anteil der Jacobi-Matrizen kann für die Drehgelenke der seriellen Struktur wie folgt berechnet werden:

$${}^i\mathbf{J}_{T,rel,i} = \frac{\partial {}^i\mathbf{v}_{v(i)i}}{\partial \dot{\mathbf{q}}} = {}^i\mathbf{A}_{v(i)} \underbrace{\left({}^{v(i)}\tilde{\mathbf{r}}_{v(i)i}^T {}^{v(i)}\mathbf{J}_{R,v(i)} \right)}_{\frac{{}^{v(i)}\mathbf{v}_{v(i)i}}{\partial \dot{\mathbf{q}}}}, \quad (3.16)$$

$${}^i\mathbf{J}_{R,\omega,rel,i} = \frac{\partial {}^i\boldsymbol{\omega}_{v(i)i}}{\partial \dot{\mathbf{q}}}, \quad (3.17)$$

mit

$$\tilde{\mathbf{r}} = \begin{pmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{pmatrix}, \quad (3.18)$$

$${}^i\boldsymbol{\omega}_{v(i)i} = \mathbf{e}_z \dot{\theta}_i = \mathbf{e}_z \dot{q}_i, \quad (3.19)$$

$${}^{v(i)}\mathbf{v}_{v(i)i} = {}^{v(i)}\boldsymbol{\omega}_{v(i)i} \times {}^{v(i)}\mathbf{r}_{v(i)i} = {}^{v(i)}\tilde{\mathbf{r}}_{v(i)i}^T {}^{v(i)}\boldsymbol{\omega}_{v(i)i}. \quad (3.20)$$

Die Winkelgeschwindigkeit ${}^i\boldsymbol{\omega}_{v(i)i}$ ergibt sich aus der Ableitung nach dem DH-Parameter θ_i . Durch $\tilde{\mathbf{r}}$ wird der Tilde Operator beschrieben, welcher eine schiefsymmetrische Matrix darstellt, die anstelle eines Kreuzprodukts verwendet werden kann [52]. Für die relative Jacobi-Matrix der Rotation ${}^i\mathbf{J}_{R,\omega,rel,i}$ erhält man eine Nullmatrix, mit einer 1 in der i -ten Spalte der dritten Zeile.

3.3. Parallele Struktur

Dieser Abschnitt wurde in Anlehnung an die Arbeit von Ellensohn [18] erstellt. Hier wird die Kinematik für die parallele Struktur der 8-Achs Bewegungsplattform berechnet.

In Abbildung 8 ist das Schema des Hexapods dargestellt, der in dieser Arbeit modelliert wird.

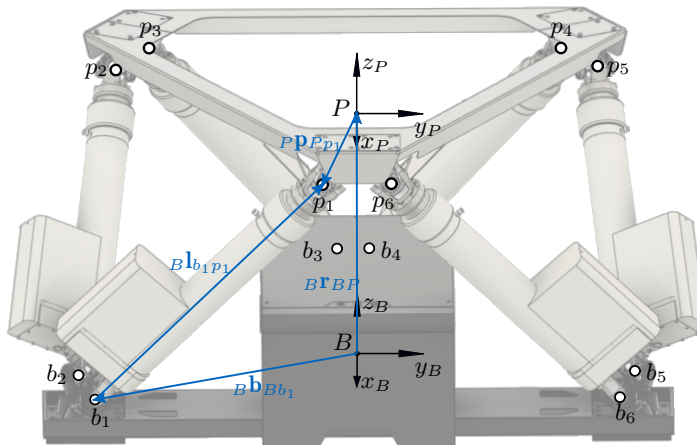


Abbildung 8 Schema des Hexapods.

Darin ist in blau die geschlossene kinematische Kette für den ersten Aktuator eingezeichnet. Weiter sind die Koordinatensysteme der unteren Plattform (B) und der oberen beweglichen Plattform (P) dargestellt sowie die Befestigungspunkte der Aktuatoren p_i und b_i an den beiden Plattformen. Für die direkte und inverse Kinematik muss jede geschlossene kinematische Kette der parallelen Struktur als geschlossene Vektorkette geschrieben werden. Daraus können für einen Hexapod sechs Vektoren berechnet werden, welche die Richtung und die Länge der Aktuatoren beschreiben:

$${}^B\mathbf{l}_{b_i p_i} = {}^B\mathbf{r}_{BP} - {}^B\mathbf{b}_{Bb_i} + {}^B\mathbf{A}_P(\boldsymbol{\beta}_p) {}^P\mathbf{p}_{Pp_i} \quad \text{mit } i = 1, \dots, 6. \quad (3.21)$$

Im Gegensatz zu seriellen Strukturen kann die inverse Kinematik von parallelen Strukturen analytisch berechnet werden. Die direkte Kinematik kann jedoch im Allgemeinen nur iterativ gelöst werden.

3.3.1. Inverse Kinematik

Aufgabe der inversen Kinematik ist die Umrechnung von Arbeitsraumkoordinaten auf Minimalkoordinaten. Dafür wird eine Funktion der Form $\mathbf{q}_p = f^{-1}(\mathbf{w}_p)$ gesucht. Die Arbeitsraumkoordinaten für den Hexapod sind folgendermaßen definiert:

$$\mathbf{w}_p = \begin{pmatrix} {}^B\mathbf{r}_{BP} \\ \mathbf{Q}_p \end{pmatrix}. \quad (3.22)$$

$\mathbf{Q}_p \in \mathbb{H}$ beschreibt dabei die Darstellung der Rotationsparameter als Quaternion. Für die Berechnung der Minimalkoordinaten des Hexapods $\mathbf{q}_p \in \mathbb{R}^6$, kann direkt Gleichung (3.21) verwendet werden. Hierfür wird die Länge von ${}^B\mathbf{l}_{b_i p_i} \in \mathbb{R}^3$ berechnet und anschließend die Ausgangslänge des Aktuators l_0 subtrahiert. Die Länge kann mithilfe von

$$l_i = f_i^{-1}(\mathbf{w}_p) = \|{}^B\mathbf{l}_{b_i p_i}\| = \sqrt{{}^B\mathbf{l}_{b_i p_i}^T {}^B\mathbf{l}_{b_i p_i}} \quad (3.23)$$

bestimmt werden. Durch Subtraktion von l_0 resultieren die Minimalkoordinaten

$$q_{p,i} = l_i - l_0 \quad (3.24)$$

der einzelnen Aktuatoren. Zur Berechnung der inversen Kinematik der gesamten 8-Achs Bewegungsplattform wird die Jacobi-Matrix benötigt, welche aus der Ableitung der Minimalkoordinaten nach der Zeit bestimmt wird. Es werden jedoch nicht direkt die Minimalkoordinaten aus (3.24) abgeleitet, sondern stattdessen die Ableitung von ${}^B\mathbf{l}_{b_i p_i}$ nach der Zeit gebildet. Das Ergebnis wird anschließend mit dem Einheitsvektor ${}^B\mathbf{n}_i \in \mathbb{R}^3$ in Richtung des Aktuators projiziert. Die Ableitung von ${}^B\mathbf{l}_{b_i p_i}$ nach der Zeit berechnet sich folgendermaßen:

$$\begin{aligned} {}^B\dot{\mathbf{l}}_{b_i p_i} &= {}^B\dot{\mathbf{r}}_{BP} + {}^B\dot{\mathbf{A}}_P {}^P\mathbf{p}_{Pp_i} \\ &= {}^B\dot{\mathbf{r}}_{BP} + \underbrace{{}^B\dot{\mathbf{A}}_P {}^P\mathbf{A}_B}_{{}^B\tilde{\boldsymbol{\omega}}} {}^B\mathbf{A}_P {}^P\mathbf{p}_{Pp_i} \\ &= {}^B\dot{\mathbf{r}}_{BP} + {}^B\tilde{\boldsymbol{\omega}} {}^B\mathbf{A}_P {}^P\mathbf{p}_{Pp_i} \\ &= {}^B\dot{\mathbf{r}}_{BP} + {}^B\boldsymbol{\omega} \times ({}^B\mathbf{A}_P {}^P\mathbf{p}_{Pp_i}). \end{aligned} \quad (3.25)$$

Durch Projektion der Ableitung in Richtung des Aktuators

$$B\mathbf{n}_i = \frac{B\dot{\mathbf{l}}_{b_i p_i}}{l_i}. \quad (3.26)$$

ergibt sich die Ableitung der Minimalkoordinaten nach der Zeit zu

$$\begin{aligned} \dot{q}_{p,i} &= B\mathbf{n}_i \cdot B\dot{\mathbf{l}}_{b_i p_i} \\ &= B\mathbf{n}_i \cdot B\dot{\mathbf{r}}_{BP} + B\mathbf{n}_i \cdot (B\boldsymbol{\omega} \times (B\mathbf{A}_{PPP} \mathbf{p}_{p_i})). \end{aligned} \quad (3.27)$$

Durch Umstellen des Spatprodukts im letzten Term resultiert²

$$\dot{q}_{p,i} = B\mathbf{n}_i \cdot B\dot{\mathbf{r}}_{BP} + (B\mathbf{A}_{PPP} \mathbf{p}_{p_i} \times B\mathbf{n}_i) \cdot B\boldsymbol{\omega}, \quad (3.28)$$

woraus sich durch Darstellung in Vektor/Matrix Schreibweise die inverse Jacobi-Matrix ergibt:

$$\dot{q}_{p,i} = \begin{pmatrix} B\mathbf{n}_i^T & (B\mathbf{p}_{p_i} \times B\mathbf{n}_i)^T \end{pmatrix} \begin{pmatrix} B\dot{\mathbf{r}}_{BP} \\ B\boldsymbol{\omega} \end{pmatrix}. \quad (3.29)$$

Für den gesamten Hexapod ist diese wie folgt definiert:

$$\dot{\mathbf{q}}_p = \underbrace{\begin{pmatrix} B\mathbf{n}_1^T & (B\mathbf{p}_{p_1} \times B\mathbf{n}_1)^T \\ B\mathbf{n}_2^T & (B\mathbf{p}_{p_2} \times B\mathbf{n}_2)^T \\ \vdots & \vdots \\ B\mathbf{n}_6^T & (B\mathbf{p}_{p_6} \times B\mathbf{n}_6)^T \end{pmatrix}}_{\mathbf{J}_{Hex,\omega}^{-1}} \begin{pmatrix} B\dot{\mathbf{r}}_{BP} \\ B\boldsymbol{\omega} \end{pmatrix}. \quad (3.30)$$

Da die Jacobi-Matrix aus der inversen Kinematik berechnet wird, erhält man als Ergebnis auch die inverse Jacobi-Matrix des Hexapods $\mathbf{J}_{Hex,\omega}^{-1}$. Diese bezieht sich auf die Winkelgeschwindigkeit und stellt deshalb die geometrische Jacobi-Matrix dar. Für die direkte Kinematik wird jedoch eine Jacobi-Matrix benötigt, welche sich auf die Quaternionen aus den Arbeitsraumkoordinaten des Hexapods \mathbf{w}_p bezieht. Folglich muss ein Zusammenhang zwischen der Winkelgeschwindigkeit $\boldsymbol{\omega}$ und der Ableitung der Quaternionen $\dot{\mathbf{Q}}$ hergestellt werden. Dieser ist durch

$$\boldsymbol{\omega} = \mathbf{T}(\mathbf{Q})\dot{\mathbf{Q}} \quad (3.31)$$

gegeben. Für die Definition der Matrix $\mathbf{T}(\mathbf{Q})$ sei auf [26] verwiesen. Die Jacobi-Matrix bezogen auf Quaternionen $\mathbf{J}_{Hex,Q}$ ergibt sich aus:

$$\mathbf{J}_{Hex,Q} = \begin{pmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-1}(\mathbf{Q}) \end{pmatrix} \mathbf{J}_{Hex,\omega}. \quad (3.32)$$

² Mithilfe von $(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$ und $(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c} = (\mathbf{b} \times \mathbf{c}) \cdot \mathbf{a} = (\mathbf{c} \times \mathbf{a}) \cdot \mathbf{b}$

Darin stellt $\mathbf{E} \in \mathbb{R}^{3 \times 3}$ eine Einheitsmatrix und $\mathbf{0} \in \mathbb{R}^{3 \times 3}$ eine Nullmatrix dar.

Die geometrische Jacobi Matrix der seriellen Struktur aus (3.13) kann gleich wie in (3.32) zu einer Jacobi-Matrix bezogen auf Quaternionen transformiert werden.

3.3.2. Direkte Kinematik

Im Gegensatz zur inversen Kinematik kann die direkte Kinematik eines Hexapods nicht analytisch gelöst werden, da sich Gleichung (3.21) nicht in der Form $\mathbf{w}_p = f(\mathbf{q}_p)$ darstellen lässt. Tatsächlich besitzt die direkte Kinematik keine eindeutige Lösung, sondern es gibt 40 gültige Lösungen [15, 28]. Für die Berechnung wird daher ein iteratives Verfahren wie das Newtonverfahren eingesetzt. Durch das Newtonverfahren kann die Nullstelle einer nichtlinearen Funktion näherungsweise bestimmt werden. Die Grundidee des Verfahrens besteht darin, die Funktion an einem Startwert zu linearisieren und den Nulldurchgang der Linearisierung als Approximation der Nullstelle der Funktion zu verwenden. In einem weiteren Iterationsschritt wird die Approximation als Startwert für die nächste Linearisierung verwendet. Dieser Vorgang wird so lange wiederholt, bis ein gewisses Abbruchkriterium erfüllt ist. Für eine detaillierte Erklärung sei auf [14] verwiesen.

Um die direkte Kinematik des Hexapods zu berechnen, wird die Nullstelle der Funktion

$$\mathbf{F}(\mathbf{w}_p^k) = \mathbf{q}_p(\mathbf{w}_p^k) - \mathbf{q}_d \quad (3.33)$$

angenähert. Darin werden die Arbeitsraumkoordinaten des Hexapods \mathbf{w}_p^k durch das Newtonverfahren so lange iteriert, bis eine Nullstelle gefunden wird. Der hochgestellte Index k beschreibt den Iterationsschritt des Newtonverfahrens. Die Funktion $\mathbf{q}_p(\mathbf{w}_p^k)$ stellt die inverse Kinematik des Hexapods dar (siehe (3.24)) und die Variable $\mathbf{q}_d \in \mathbb{R}^6$ die gewünschten Aktuatorpositionen, für die eine Lösung der direkten Kinematik gefunden werden soll. Durch den Index d wird immer signalisiert, dass es sich um einen gewünschten (engl. desired) bzw. vorgegebenen Wert handelt.

Die Iteration wird so lange durchgeführt, bis eines der beiden Abbruchkriterien erfüllt ist:

$$\|\mathbf{F}(\mathbf{w}_p^k)\| < \epsilon_1, \quad \text{oder} \quad \|\mathbf{w}_p^k - \mathbf{w}_p^{k-1}\| < \epsilon_2. \quad (3.34)$$

Um die Arbeitsraumkoordinaten für den nächsten Zeitschritt zu berechnen, wird im Newtonverfahren folgende Formel verwendet:

$$\mathbf{w}_p^{k+1} = \mathbf{w}_p^k - \frac{\mathbf{F}(\mathbf{w}_p^k)}{\mathbf{F}'(\mathbf{w}_p^k)} \quad (3.35)$$

$$= \mathbf{w}_p^k - \left(\frac{\partial \mathbf{F}(\mathbf{w}_p^k)}{\partial \mathbf{w}_p^k} \right)^{-1} \mathbf{F}(\mathbf{w}_p^k) \quad (3.36)$$

$$= \mathbf{w}_p^k - \mathbf{J}_{Hex,Q}(\mathbf{w}_p^k) \mathbf{F}(\mathbf{w}_p^k). \quad (3.37)$$

Dabei beschreibt der Ausdruck $\frac{\partial \mathbf{F}(\mathbf{w}_p^k)}{\partial \mathbf{w}_p^k}$ die inverse Jacobi-Matrix der Arbeitsraumkoordinaten des Hexapods aus (3.32).

Die Lösung der Arbeitsraumkoordinaten \mathbf{w}_p^k (aus dem finalen Iterationsschritt) stellt die Lösung der direkten Kinematik dar, welche aus den gewünschten Aktuatorpositionen \mathbf{q}_d ermittelt wurde.

Ein entscheidender Faktor bei der iterativen Berechnung mit dem Newtonverfahren ist die Wahl des Startwerts. Da die Steuerung des Hexapods so aufgebaut ist, dass sich dieser nach dem Einschalten immer aus einer fest definierten Ausgangsstellung wegbewegt, kann diese Ausgangsstellung als Startwert verwendet werden. Wird eine Trajektorie abgefahren, dient immer die Lösung aus dem letzten Zeitschritt als Startwert für die nächste Iteration. In allen durchgeführten Simulationen konnte so ein gutes Konvergenzverhalten des Hexapods beobachtet werden.

3.4. Hybride Struktur

In diesem Abschnitt wird die Kinematik des seriellen und parallelen Teils der 8-Achs Bewegungsplattform miteinander verbunden. Eine Übersicht des gesamten Systems ist in Abbildung 5 und Abbildung 6 dargestellt.

Die Kinematik von hybriden Strukturen kann unter einer Bedingung auf die gleiche Weise berechnet werden wie die Kinematik von seriellen Strukturen. Dafür müssen die parallelen Anteile des Gesamtsystems als einzelne Körper betrachtet werden, auf denen ein Gelenk befestigt ist, welches die gleichen Freiheitsgrade wie die jeweilige parallele Struktur besitzt. So kann die Kinematik der parallelen Teile als relativkinematische Größe für die rekursive Berechnung der Kinematik der hybriden Struktur genutzt werden.

Die Minimalkoordinaten $\mathbf{q}_h = (\mathbf{q}_p, \mathbf{q}_s) \in \mathbb{R}^8$ der gesamten 8-Achs Bewegungsplattform setzen sich aus den Minimalkoordinaten \mathbf{q}_p des Hexapods und den Minimalkoordinaten \mathbf{q}_s der seriellen Plattform zusammen.

3.4.1. Direkte Kinematik

Die direkte Kinematik des Gesamtsystems wird durch die Aneinanderreihung von homogenen Transformation ermittelt. Für den Hexapod wird die homogene Transformation aus den Arbeitsraumkoordinaten \mathbf{w}_p^k erstellt:

$${}^B\mathbf{T}_P = \begin{pmatrix} \mathbf{A}(\mathbf{Q}_p)^T & {}^B\mathbf{r}_{BP} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} {}^B\mathbf{A}_P & {}^B\mathbf{r}_{BP} \\ \mathbf{0}^T & 1 \end{pmatrix}. \quad (3.38)$$

Diese wird mit der homogenen Transformation der seriellen Struktur ${}^I\mathbf{T}_T$ aus (3.9) multipliziert. Das Koordinatensystem I aus Abbildung 7 wird dafür durch das Koordinatensystem P der beweglichen Plattform des Hexapods ersetzt. So erhält man ${}^P\mathbf{T}_T$ anstatt ${}^I\mathbf{T}_T$ und folglich:

$${}^B\mathbf{T}_T = {}^B\mathbf{T}_{PP}{}^P\mathbf{T}_T = \begin{pmatrix} {}^B\mathbf{A}_T & {}^B\mathbf{r}_{BT} \\ \mathbf{0}^T & 1 \end{pmatrix}. \quad (3.39)$$

Im Anschluss können daraus die Arbeitsraumkoordinaten der 8-Achs Bewegungsplattform $\mathbf{w}_h \in \mathbb{R}^6$ berechnet werden:

$$\mathbf{w}_h = \begin{pmatrix} {}^B\mathbf{r}_{BT} \\ \mathbf{Q}({}^T\mathbf{A}_B) \end{pmatrix} = \begin{pmatrix} \mathbf{r}_h \\ \mathbf{Q}_h \end{pmatrix}. \quad (3.40)$$

3.4.2. Inverse Kinematik

Wie in Abschnitt 3.2.2 schon erklärt wird die inverse Kinematik der Gesamtstruktur auf Geschwindigkeitsebene gelöst. Dafür wird die geometrische Jacobi-Matrix der gesamten 8-Achs Bewegungsplattform benötigt.

Durch Invertieren der in (3.30) berechneten inversen Jacobi-Matrix des Hexapods und anschließendem Hinzufügen von zwei Nullspalten auf der rechten Seite, resultiert die relative Jacobi-Matrix ${}_B\mathbf{J}_{g,P} \in \mathbb{R}^{6 \times 8}$. Diese wird mithilfe der rekursiven Berechnungsweise (siehe (3.15) und (3.14)) durch die Jacobi-Matrix der seriellen Struktur erweitert. Es ergibt sich dadurch die geometrische Jacobi-Matrix ${}_B\mathbf{J}_{g,T} \in \mathbb{R}^{6 \times 8}$ der gesamte Bewegungsplattform, welche in (3.12) eingesetzt die Basis zur Lösung der Minimalkoordinaten \mathbf{q}_h darstellt. Da es sich hier um ein redundantes System handelt, kann die Jacobi-Matrix nicht einfach invertiert werden, um die Ableitung der Minimalkoordinaten nach der Zeit $\dot{\mathbf{q}}_h$ zu berechnen, weil es in der Regel unendlich viele Lösungen gibt. Um dennoch eine Lösung zu erhalten, gibt es Verfahren, denen ein Optimierungsproblem auf Geschwindigkeitsebene zugrunde liegt [38]. Hierbei werden zu jedem Zeitpunkt lokal die Gelenkgeschwindigkeiten $\dot{\mathbf{q}}_h$ minimiert.

Im Folgenden werden Verfahren zur Berechnung der inversen Kinematik vorgestellt. Diese werden allgemein beschrieben und nicht spezifisch für die Bewegungsplattform dargestellt, weshalb auch die Variablen in den Formeln keine Indizes für Koordinatensysteme oder Strukturen besitzen.

Ein mögliches Verfahren stellt das „Resolved Motion Rate Control“ [56] dar:

$$\dot{\mathbf{q}} = \underbrace{\mathbf{W}^{-1} \mathbf{J}_g^T (\mathbf{J}_g \mathbf{W}^{-1} \mathbf{J}_g^T)^{-1}}_{:= \mathbf{J}_g^\dagger} \dot{\mathbf{x}}_d. \quad (3.41)$$

Hierbei beschreibt $\mathbf{J}_g^\dagger \in \mathbb{R}^{n \times 6}$ die mit $\mathbf{W} \in \mathbb{R}^{n \times n}$ gewichtete Moore-Penrose Pseudoinverse der geometrischen Jacobi-Matrix $\mathbf{J}_g \in \mathbb{R}^{6 \times n}$. Die symmetrische und positiv definite Matrix \mathbf{W} , welche meist als Diagonalmatrix definiert wird, erlaubt eine Gewichtung der einzelnen Gelenke. Wird der Wert eines diagonalen Eintrags vergrößert, verringert sich die Geschwindigkeit des zugehörigen Gelenks, was einer niedrigeren Gewichtung entspricht. Des Weiteren stellt die Variable $\dot{\mathbf{x}}_d \in \mathbb{R}^6$ die gewünschten Geschwindigkeiten im kartesischen Raum dar.

Eine Erweiterung des „Resolved Motion Rate Control“ ist das „Automatic Supervisory Control“ [2]:

$$\dot{\mathbf{q}} = \mathbf{J}_g^\dagger \dot{\mathbf{x}}_d + \underbrace{(\mathbf{E} - \mathbf{J}_g^\dagger \mathbf{J}_g)}_{\mathbf{N}_g} \mathbf{W}^{-1} \dot{\mathbf{q}}_0. \quad (3.42)$$

Dabei können nicht nur die einzelnen Gelenke über die Matrix \mathbf{W} gewichtet werden, sondern es wird zusätzlich eine beliebige Nullraumgeschwindigkeit $\dot{\mathbf{q}}_0 \in \mathbb{R}^n$ mit der Nullraum-Projektionsmatrix $\mathbf{N}_g \in \mathbb{R}^{n \times n}$ in den Nullraum projiziert. Darin beschreibt $\mathbf{E} \in \mathbb{R}^{n \times n}$ die Einheitsmatrix. Bei einem redundanten mechanischen System ist es in der Regel möglich, die Minimalkoordinaten \mathbf{q} zu variieren, ohne dabei die Pose des Endeffektors zu verändern. Es wird daher von einer Bewegung im Nullraum gesprochen. Das bedeutet, dass $\dot{\mathbf{q}}_0$ die vorgegebene Geschwindigkeit des Endeffektors nicht beeinflusst, sondern nur die Bewegung im Nullraum.

Nullraumgeschwindigkeit

Die Nullraumgeschwindigkeit $\dot{\mathbf{q}}_0$ kann auf unterschiedliche Weise eingesetzt werden. Beispielsweise zur Vorgabe einer bevorzugten Konfiguration des Systems, zur Kollisionsvermeidung oder zur Beschränkung der Gelenke [38]. Es wird dabei jedoch nicht sichergestellt, dass die Vorgabe von $\dot{\mathbf{q}}_0$ eingehalten wird. Tritt zum Beispiel eine Singularität auf, kommt es zum Rang-Abfall der Jacobi-Matrix und der Nullraum wird verkleinert oder verschwindet ganz. Folglich kann es dazu kommen, dass die Nullraumgeschwindigkeit $\dot{\mathbf{q}}_0$ ignoriert wird. Weiter ist es möglich, dass die Geschwindigkeit aus dem ersten Term von (3.42) viel größer ist als die Geschwindigkeit des zweiten Terms. In diesem Fall kann es auch dazu kommen, dass Beschränkungen nicht eingehalten werden. Dieses Problem lässt sich in manchen Fällen durch eine Modifizierung der Gewichtung lösen.

Im Rahmen dieser Arbeit wird $\dot{\mathbf{q}}_0$ verwendet, um die Position und Geschwindigkeit zu beschränken. Dafür wird

$$\dot{\mathbf{q}}_0 = \alpha_{pos} \mathbf{H}_{pos}(\mathbf{q}) + \alpha_{vel} \mathbf{H}_{vel}(\dot{\mathbf{q}}) \quad (3.43)$$

gesetzt. Die α -Terme stellen einen skalaren Gewichtungsfaktor für die Funktionen der Positionsbeschränkung $\mathbf{H}_{pos}(\mathbf{q}) \in \mathbb{R}^8$ und der Geschwindigkeitsbeschränkung $\mathbf{H}_{vel}(\dot{\mathbf{q}}) \in \mathbb{R}^8$ dar. Diese sind für $i = 1, \dots, 8$, wie folgt definiert:

$$H_{pos,i}(q_i) = \begin{cases} \left(\frac{q_i - q_{min,s,i}}{q_{min,s,i} - q_{min,i}} \right)^2 & \text{für } q_i \leq q_{min,s,i} \\ - \left(\frac{q_i - q_{max,s,i}}{q_{max,i} - q_{max,s,i}} \right)^2 & \text{für } q_i \geq q_{max,s,i} \\ 0 & \text{sonst } \wedge i = 7 \wedge i = 8 \end{cases}, \quad (3.44)$$

$$H_{vel,i}(\dot{q}) = \begin{cases} \left(\frac{\dot{q}_i - \dot{q}_{min,s,i}}{\dot{q}_{min,s,i} - \dot{q}_{min,i}} \right)^2 & \text{für } \dot{q}_i \leq \dot{q}_{min,s,i} \\ - \left(\frac{\dot{q}_i - \dot{q}_{max,s,i}}{\dot{q}_{max,i} - \dot{q}_{max,s,i}} \right)^2 & \text{für } \dot{q}_i \geq \dot{q}_{max,s,i} \\ 0 & \text{sonst} \end{cases}, \quad (3.45)$$

mit:

$$\begin{aligned} q_{min,s,i} &= q_{min,i} + q_{eps,i}, \\ q_{max,s,i} &= q_{max,i} - q_{eps,i}, \\ \dot{q}_{min,s,i} &= \dot{q}_{min,i} + \dot{q}_{eps,i}, \\ \dot{q}_{max,s,i} &= \dot{q}_{max,i} - \dot{q}_{eps,i}. \end{aligned}$$

Die Funktion der Positionsbeschränkung $\mathbf{H}_{pos}(\mathbf{q})$ wird durch die unteren Antriebsbeschränkungen $\mathbf{q}_{min} \in \mathbb{R}^8$ und oberen Antriebsbeschränkungen $\mathbf{q}_{max} \in \mathbb{R}^8$, sowie den Grenzwerten $\mathbf{q}_{eps} \in \mathbb{R}^8$, welche festlegen, ab wann ein Gelenk zu bremsen beginnt, definiert. Analog dazu wird die Geschwindigkeitsbeschränkung $\mathbf{H}_{vel}(\dot{\mathbf{q}})$ durch die unteren Geschwindigkeitslimits $\dot{\mathbf{q}}_{min} \in \mathbb{R}^8$ und die oberen Geschwindigkeitslimits $\dot{\mathbf{q}}_{max} \in \mathbb{R}^8$ sowie den Grenzwerten $\dot{\mathbf{q}}_{eps} \in \mathbb{R}^8$ beschrieben. In (3.44) und (3.45) ist eine Funktion definiert, die gleich null ist, wenn sich die Position und Geschwindigkeit unter einem gewissen Grenzwert ($q_{min,s,i}, q_{max,s,i}, \dot{q}_{min,s,i}, \dot{q}_{max,s,i}$) befinden. Erst wenn dieser Wert

überschritten wird, steigt der Funktionswert quadratisch an und erreicht am eingestellten Limit den Wert eins. Die Konfiguration der Bewegungsplattform wird dadurch erst beeinflusst, wenn sich ein Gelenk kurz vor dem Erreichen der maximalen oder minimalen Position bzw. Geschwindigkeit befindet. In den Abbildungen 9a und 9b ist die Nullraumgeschwindigkeit $\dot{q}_{0,i}$ für ein beispielhaftes Gelenk abgebildet. Wie in Abbildung 9a zu erkennen ist, entsteht erst kurz vor den Gelenklimits bei $q_{min} = 0$ und $q_{max} = 0.6$ eine Nullraumgeschwindigkeit, welche das Gelenk vom Limit wegdrückt.

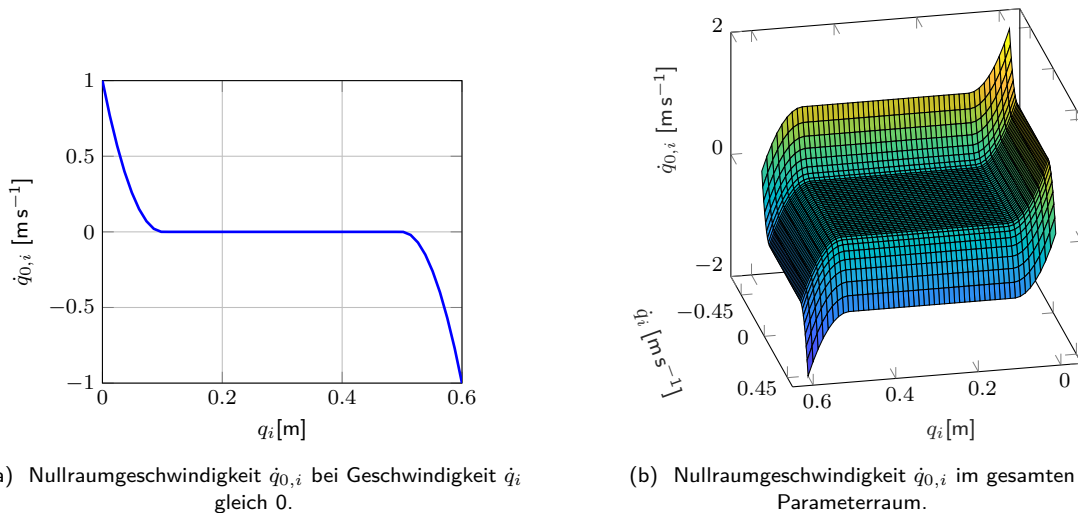


Abbildung 9 Nullraumgeschwindigkeit $\dot{q}_{0,i}$ für ein Gelenk i mit den Parameterwerten $\alpha_{pos,i} = 1$, $\alpha_{vel,i} = 1$, $q_{min,i} = 0$, $q_{max,i} = 0.6$, $q_{eps,i} = 0.1$, $\dot{q}_{min,i} = -0.45$, $\dot{q}_{max,i} = 0.45$ und $\dot{q}_{eps,i} = 0.15$.

3.5. Inverse Kinematik mit Prioritäten

Für redundante mechanische Systeme besteht die Möglichkeit, mehrere Aufgaben gleichzeitig auszuführen, welche unterschiedliche Prioritäten besitzen. Diese Priorisierung wird durch Nullraumprojektionen hergestellt. Das „Automatic Supervisory Control“ (siehe (3.42)) stellt die Grundlage für diese Verfahren dar. Durch die Projektion von \dot{q}_0 in den Nullraum kann die Aufgabe, welche es ist die gewünschte Trajektorie abzufahren, nicht von \dot{q}_0 beeinflusst werden [27].

In [39] wurde ein Verfahren entwickelt, das es ermöglicht, zwei Aufgaben zu definieren, die unterschiedliche Prioritäten besitzen. Diese Methode wurde in [50] für eine beliebige Anzahl an Aufgaben erweitert. Ein Problem dabei ist, dass algorithmische Singularitäten (engl. algorithmic singularity), welche nicht aufgrund der kinematischen Konfiguration des Systems auftreten, sondern aufgrund der Art der Berechnung der inversen Kinematik. Ein Lösungsansatz zur Vermeidung der algorithmischen Singularitäten wurde in [10] durch das Umstellen der Gleichungen auf das sogenannte SRMTP (engl. singularity-robust multiple task priority) Framework entwickelt. Auch dieses wurde auf ein Verfahren für eine beliebige Anzahl an Aufgaben erweitert, das unter dem Namen nullraumbasierte Verhaltenssteuerung (engl. Null Space-based Behavioral control, kurz: NSB control) bekannt ist [3]. In [44] wurden beide Formulierungen miteinander verglichen, mit dem Ergebnis, dass im SRMTP framework und bei der NSB Steuerung zwar keine algorithmischen Singularitäten auftreten, jedoch die Aufgaben mit niedrigem Prioritätslevel nicht so genau eingehalten werden, wie im ursprünglichen

Verfahren aus [50].

Die Bewegungsplattform besitzt im Arbeitsraum keine kinematische Singularität. Es wird deshalb auch keine spezielle Methode zur Berechnung der Moore-Penrose Pseudoinversen benötigt, wie zum Beispiel das „Damped Least-Squares“ Verfahren [13]. In dieser Arbeit wird folglich die NSB Steuerung eingesetzt, da dabei keine algorithmischen Singularitäten entstehen und so Singularitäten nicht weiter beachtet werden müssen.

Durch die NSB Steuerung können k verschiedene Aufgaben $\sigma_i = \mathbf{f}_{\sigma_i}(\mathbf{q}) \in \mathbb{R}^{m_i}$ (mit $i = 1, \dots, k$) definiert werden, wobei σ_1 die höchste Priorität und σ_k die niedrigste Priorität besitzt. Eine Aufgabe kann zum Beispiel das Abfahren einer Trajektorie des Endeffektors sein.

Der differentielle Zusammenhang zwischen der Aufgabe σ_i und der Minimalkoordinaten \mathbf{q} ist durch

$$\dot{\sigma}_i = \frac{\partial \mathbf{f}_{\sigma_i}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_i(\mathbf{q}) \dot{\mathbf{q}} \quad (3.46)$$

gegeben, worin $\mathbf{J}_i \in \mathbb{R}^{m_i \times n}$ die Jacobi-Matrix der Aufgabe beschreibt.

Mithilfe des differentiellen Zusammenhangs lassen sich für die NSB Steuerung die Gelenkgeschwindigkeiten

$$\dot{\mathbf{q}}_{NSB} = \dot{\mathbf{q}}_1 + \mathbf{N}_{1,1} \dot{\mathbf{q}}_2 + \dots + \mathbf{N}_{1,k-1} \dot{\mathbf{q}}_k, \quad (3.47)$$

berechnen. Die Geschwindigkeit $\dot{\mathbf{q}}_i \in \mathbb{R}^n$ beschreibt dabei die Geschwindigkeit der einzelnen Aufgaben, welche im Normalfall durch $\dot{\mathbf{q}}_i = \mathbf{J}_i^\dagger \dot{\sigma}_i$ berechnet werden. Für den letzten Geschwindigkeitsvektor $\dot{\mathbf{q}}_k$ ist es möglich, einen beliebigen Geschwindigkeitsvektor zu verwenden, wie zum Beispiel die Nullraumgeschwindigkeit $\dot{\mathbf{q}}_0$ aus (3.43). Die Moore-Penrose Pseudoinverse \mathbf{J}_i^\dagger (siehe (3.41)) wird aus der Jacobi-Matrix \mathbf{J}_i jeder Aufgabe bestimmt. Zusätzlich wird diese für die erweiterte Jacobi-Matrix (engl. augmented Jacobian) benötigt:

$$\mathbf{J}_{1,k} = \begin{pmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \vdots \\ \mathbf{J}_k \end{pmatrix}. \quad (3.48)$$

Daraus wird wiederum die Nullraumprojektionsmatrix

$$\mathbf{N}_{1,i} = \left(\mathbf{E} - \mathbf{J}_{1,i}^\dagger \mathbf{J}_{1,i} \right), \quad (3.49)$$

gebildet, worin $\mathbf{E} \in \mathbb{R}^{n \times n}$ die Einheitsmatrix darstellt. So wird ausgehend von Aufgabe σ_1 jede weitere Aufgabe in den Nullraum der höher priorisierten Aufgaben projiziert. Das bedeutet, dass die niedriger priorisierten Aufgaben nur ausgeführt werden, wenn diese die höher priorisierten Aufgaben nicht beeinflussen.

In der NSB Steuerung können nur Aufgaben definiert werden, die eine Gleichheitsbedingung erfüllen (engl. equality task). Das bedeutet, dass beispielsweise eine gewünschte Trajektorie $\sigma(t) = \sigma_d(t)$

vorgegeben wird. Oft ist es jedoch nützlich, auch Aufgaben zu definieren, die eine Ungleichheitsbedingung erfüllen (engl. set-based task). Ziel ist hierbei, die Aufgabe $\sigma(t) \in D \forall t$ in einer definierten Menge D (grüner Bereich in Abbildung 10) zu halten, anstatt diese auf einen festen Wert zu steuern. Ein „set-based task“ kann als ein „equality task“ gesehen werden, der je nachdem ob eine Bedingung erfüllt ist oder nicht, aktiviert oder deaktiviert wird. Beispiele dafür sind die Einhaltung von Aktuatorbegrenzungen oder eine Kollisionsvermeidung.

In [35] wurde ein Verfahren entwickelt, das es ermöglicht, skalare Aufgaben σ_i zu definieren, welche durch eine Ungleichheitsbedingung beschrieben werden. Zur Definition einer solchen Aufgabe werden mehrere Referenzwerte festgelegt. Diese sind in Abbildung 10 grafisch dargestellt.

Die Extremwerte der Aufgabe sind durch σ_{max} und σ_{min} gegeben. Um die Einhaltung der Extremwerte bei etwaigen kleinen Rechenfehlern des Verfahrens zu garantieren, werden in einem Sicherheitsabstand s die Limits $\sigma_{l,max} = \sigma_{max} - s$ und $\sigma_{l,min} = \sigma_{min} + s$ definiert. Zusätzlich werden die Grenzwerte $\sigma_{a,max} = \sigma_{l,max} - \varepsilon$ und $\sigma_{a,min} = \sigma_{l,min} + \varepsilon$ festgelegt. Bei Überschreiten eines Grenzwerts wird die Aufgabe als „equality task“ in die NSB Steuerung aus (3.47) integriert. Folgende Gleichung definiert den Wert, den diese Aufgabe dann erfüllen soll:

$$\sigma_i = \begin{cases} \sigma_{l,max} & \text{für } \sigma \geq \sigma_{a,max} \\ \sigma_{l,min} & \text{für } \sigma \leq \sigma_{a,min} \end{cases} \quad (3.50)$$

Ein Nachteil dieser Methode ist, dass durch das Hinzufügen und Entfernen von Aufgaben Geschwindigkeitssprünge in (3.47) entstehen. Diese können jedoch durch geeignete Beschleunigungsbeschränkungen oder Glättungsfunktionen [36] verringert werden.

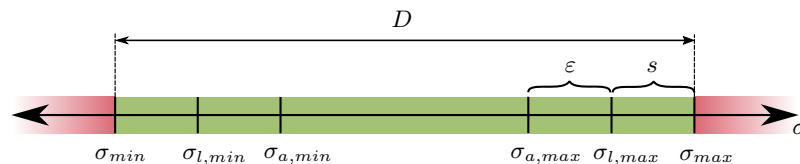


Abbildung 10 Referenzwerte für eine Aufgabe mit Ungleichungsbedingungen.

3.6. Inverse Kinematik mit Rückführung

Die inverse Kinematik wird in der Regel nicht für eine feste Konfiguration, sondern für eine ganze Trajektorie $\mathbf{w}_d(t) \in \mathbb{R}^m$ bestimmt. Um die Minimalkoordinaten $\mathbf{q}(t)$ zu berechnen, wird die Gelenkgeschwindigkeit $\dot{\mathbf{q}}$ aus Gleichung (3.42) integriert. Dabei kommt es über die Zeit zu einem numerischen Drift und bei ungenauem Startwert kann dies zu einem Positions- bzw. Orientierungsfehler führen. Zur Vermeidung solcher Fehler kann eine inverse Kinematik mit Rückführung verwendet werden. Diese wird in der Literatur als „closed-loop inverse kinematic“ (CLIK) bezeichnet [51]. Die gewünschten Arbeitsraumkoordinaten \mathbf{w}_d werden dabei mit den tatsächlichen Arbeitsraumkoordinaten $\mathbf{w}(\mathbf{q})$, welche aus der direkten Kinematik bestimmt werden, verglichen und daraus eine modifizierte

Geschwindigkeit $\dot{\mathbf{x}}_{d,mod} \in \mathbb{R}^6$ erstellt. Für den translatorischen Fehler \mathbf{e}_r ergibt sich:

$$\mathbf{e}_r = \mathbf{r}_d - \mathbf{r}(\mathbf{q}). \quad (3.51)$$

In [53] wird der rotatorische Fehler aus der Differenz der gewünschten Quaternion $\mathbf{Q}_d = (\eta_d, \boldsymbol{\epsilon}_d)$ und der gemessenen Quaternion $\mathbf{Q}(\mathbf{q}) = (\eta(\mathbf{q}), \boldsymbol{\epsilon}(\mathbf{q}))$ gebildet. Um die relative Rotation zwischen den beiden Quaternionen zu berechnen, muss das Gemessene invertiert ($\mathbf{Q}^{-1} = (\eta, -\boldsymbol{\epsilon})$) und mit der gewünschten Quaternion multipliziert werden [52]:

$$\Delta\mathbf{Q} = \begin{pmatrix} \Delta\eta \\ \Delta\boldsymbol{\epsilon} \end{pmatrix} = \mathbf{Q}_d * \mathbf{Q}^{-1}(\mathbf{q}). \quad (3.52)$$

Weisen beide Quaternionen keine relative Verdrehung auf, ergibt sich $\Delta\mathbf{Q} = (1, \mathbf{0})$. Folglich ergibt sich für den rotatorischen Fehler:

$$\mathbf{e}_\omega = \Delta\boldsymbol{\epsilon} = \eta(\mathbf{q})\boldsymbol{\epsilon}_d - \eta_d\boldsymbol{\epsilon}(\mathbf{q}) - \boldsymbol{\epsilon}_d \times \boldsymbol{\epsilon}(\mathbf{q}). \quad (3.53)$$

Das bedeutet, dass nur die Abweichung der Rotationsachse der Quaternion $\boldsymbol{\epsilon}$ in die Berechnung des rotatorischen Fehlers eingeht und $\Delta\eta$ vernachlässigt wird. Da es sich um die Rotationsachse handelt, kann damit auch direkt die Winkelgeschwindigkeit beeinflusst werden.

Die modifizierte Geschwindigkeit $\dot{\mathbf{x}}_{d,mod}$ ergibt sich folgendermaßen:

$$\dot{\mathbf{x}}_{d,mod} = \begin{pmatrix} \dot{\mathbf{r}}_d + \mathbf{K}_r \mathbf{e}_r \\ \boldsymbol{\omega}_d + \mathbf{K}_\omega \mathbf{e}_\omega \end{pmatrix}. \quad (3.54)$$

Darin beschreibt $\dot{\mathbf{r}}_d$ und $\boldsymbol{\omega}_d$ die gewünschte translatorische Geschwindigkeit und Winkelgeschwindigkeit. Die Matrizen $\mathbf{K}_r \in \mathbb{R}^{3 \times 3}$ und $\mathbf{K}_\omega \in \mathbb{R}^{3 \times 3}$ sind Diagonalmatrizen, welche einen Verstärkungsfaktor für die translatorische und rotatorische Abweichung \mathbf{e}_r und \mathbf{e}_ω darstellen. In [52] wird gezeigt, dass sich durch das Ersetzen von $\dot{\mathbf{x}}_d$ mit $\dot{\mathbf{x}}_{d,mod}$ ein konvergentes Verfahren ohne numerischen Drift ergibt.

Für ein besseres Verständnis ist in Abbildung 11 das Schema der inversen Kinematik mit Rückführung am Beispiel des „Resolved Motion Rate Control“ aus (3.41) grafisch dargestellt. Diese Rückführung kann ebenso in andere Verfahren, wie zum Beispiel das „Automatic Supervisory Control“ aus (3.42), integriert werden.

Die Eingänge für das Verfahren sind die Position $\mathbf{r}_d(t)$, die Geschwindigkeit $\dot{\mathbf{r}}_d(t)$, die Winkelgeschwindigkeit $\boldsymbol{\omega}_d(t)$ und die Orientierung in Form einer Drehmatrix $\mathbf{A}_d(t)$.

Die Winkelgeschwindigkeit ${}^B\boldsymbol{\omega}_d(t)$ wird in dieser Arbeit aus der Ableitung der Orientierung ${}^T\dot{\mathbf{A}}_{d,B}(t)$ berechnet [32]:

$${}^B\tilde{\boldsymbol{\omega}}_d(t) = {}^T\dot{\mathbf{A}}_{d,B}^T(t) {}^T\mathbf{A}_{d,B}(t) = {}^B\dot{\mathbf{A}}_{d,T}(t) {}^T\mathbf{A}_{d,B}(t) = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \quad (3.55)$$

woraus sich die Winkelgeschwindigkeit direkt ablesen lässt. Diese ist wie folgt definiert:

$${}^B\boldsymbol{\omega}_d(t) = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}. \quad (3.56)$$

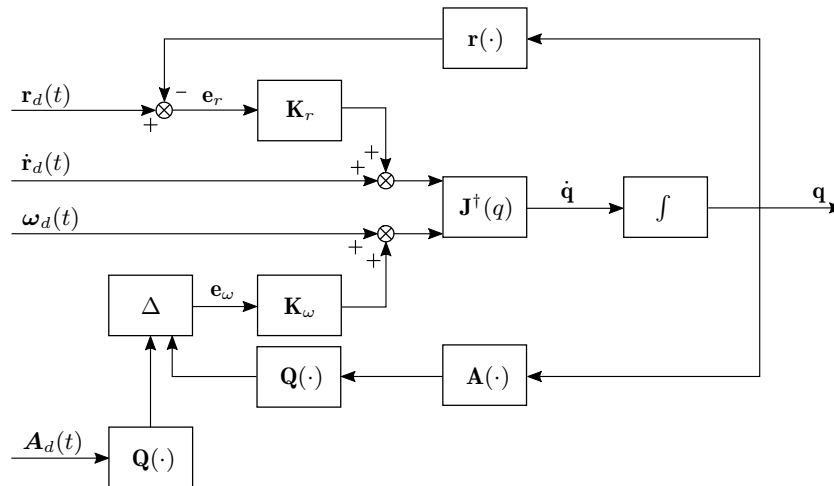


Abbildung 11 Inverse Kinematik als geschlossener Regelkreis, nach [53].

4. Gelenkbahnplanung

Aufbauend auf der inversen Kinematik der 8-Achs Bewegungsplattform wurde eine Gelenkbahnplanung entwickelt, welche in diesem Kapitel vorgestellt wird. Wie in Abbildung 12 gezeigt, stellt diese das Bindeglied zwischen einer übergeordneten kartesischen Bahnplanung und dem Steuercomputer der Bewegungsplattform dar. Die Aufgabe der Gelenkbahnplanung ist die Umrechnung einer gewünschte kartesische Trajektorie, welche aus einer Position $\mathbf{r}_d \in \mathbb{R}^3$ und einer Orientierung $\mathbf{A}_d \in \mathbb{R}^{3 \times 3}$ besteht, in Steuerbefehle \mathbf{q}_d für die einzelnen Aktuatoren der Bewegungsplattform. Hierbei müssen gleichzeitig die Positions-, Geschwindigkeits- und Beschleunigungslimits der Aktuatoren berücksichtigt werden.

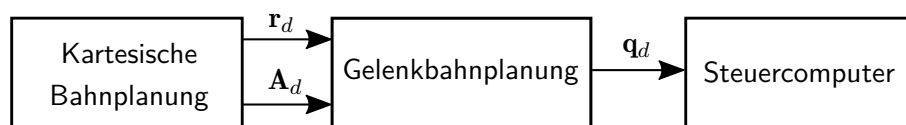


Abbildung 12 Aufbau der Steuerung für die Bewegungsplattform.

Als Grundlage für die Gelenkbahnplanung dient die NSB Steuerung, welche in Abschnitt 3.5 erklärt wurde. Ein großer Vorteil ist, dass damit die Positionsbeschränkungen direkt in die inverse Kinematik integriert werden können. Das hat im Gegensatz zu Methoden, bei denen die Positionsbeschränkungen erst auf die Lösung der inversen Kinematik angewendet werden [6, 23], den Vorteil, dass die unbeschränkten Aktuatoren zur Erfüllung der vorgegebenen Aufgabe optimal ausgenutzt werden können.

Mit der NSB Steuerung kann außerdem die Vorgabe der Trajektorie auf mehrere Aufgaben aufgeteilt werden. Das hat den Vorteil, dass bei einem Freiheitsgradverlust aufgrund von Singularitäten oder Aktuatorbeschränkungen gewisse Bewegungen höher priorisiert werden können. In der Bewegungssimulation ist es beispielsweise von Vorteil, die Orientierung des Endeffektors höher zu gewichten als die Position, da ein Orientierungsfehler als störender empfunden wird. In der Gelenkbahnplanung der Bewegungsplattform wird deshalb die Orientierung höher priorisiert, wenn Aktuatoren aufgrund einer Positionsbeschränkung gesperrt sind. Sind keine Aktuatoren gesperrt, ist eine Aufteilung der Trajektorie auf mehrere Aufgaben überflüssig, da es im Arbeitsraum keine Singularitäten gibt und deshalb auch immer genügend Freiheitsgrade zur Verfügung stehen, um die Trajektorie exakt einhalten zu können.

Nachfolgend werden in Abschnitt 4.1 die Methoden erklärt, mit denen die Positionen, Geschwindigkeiten und Beschleunigungen der Aktuatoren beschränkt werden. Darauf folgt in Abschnitt 4.2 eine Definition der benötigten Aufgaben für die NSB Steuerung. Abschließend wird in Abschnitt 4.3 die Funktionsweise der Gelenkbahnplanung vorgestellt.

4.1. Aktuatorbeschränkungen

In diesem Abschnitt wird die Beschränkung der Positionen, Geschwindigkeiten und Beschleunigungen für die Aktoren der Bewegungsplattform erläutert.

Positionsbeschränkung. Um die Position eines Aktuators mit der Bedingung

$$q_{min,i} \leq q_i \leq q_{max,i} \quad \text{für } i = 1, \dots, n \quad (4.1)$$

und unter Berücksichtigung des Beschleunigungslimits $\ddot{q}_{brake,i}$ zu begrenzen, kann eine sogenannte Bang-Bang Steuerung eingesetzt werden [5]. Dabei stellt $q_{min,i}$ und $q_{max,i}$ das minimale und maximale Positionslimit dar.

Der Bremsweg $s_{brake,i}$, den ein Aktuator im zeitdiskreten Fall benötigt, um zum Stillstand zu kommen, ohne dabei die maximale Bremsbeschleunigung $\ddot{q}_{brake,i}$ zu überschreiten, kann mit

$$s_{brake,i} = \left| \frac{\dot{q}_{old,i}^2}{2\ddot{q}_{brake,i}} \right| + |\dot{q}_{old,i}| \Delta t + \ddot{q}_{brake,i} \Delta t^2 \quad (4.2)$$

berechnet werden [5]. Dabei beschreibt $\dot{q}_{i,old}$ die Gelenkgeschwindigkeit aus dem letzten Zeitschritt und Δt die Abtastzeit. Für den zeitkontinuierlichen Fall würde der erste Term von (4.2) ausreichen, um den Aktuator genau am Limit abzubremsen. Für den zeitdiskreten Fall kann es aber passieren, dass die Bremsbedingung zwischen zwei Zeitschritten eintreten müsste, damit das Positionslimit nicht verletzt wird. Deshalb wird hier vom schlechtestmöglichen Fall ausgegangen, bei dem die Gelenkgeschwindigkeit $\dot{q}_{i,old}$ für einen vollen weiteren Zeitschritt konstant bleibt und der Aktuator maximal beschleunigt. Dadurch kann sichergestellt werden, dass der Aktuator innerhalb des berechneten Bremswegs zum Stillstand kommt.

Mit diesem Bremsweg muss in jedem Zeitschritt überprüft werden, ob ein Positionslimit verletzt wird. Dies wird durch folgende Bedingung realisiert:

$$C_i = (\dot{q}_i > 0 \quad \wedge \quad q_i + s_{brake,i} \geq q_{max,i}) \quad \vee \quad (\dot{q}_i < 0 \quad \wedge \quad q_i - s_{brake,i} \leq q_{min,i}). \quad (4.3)$$

Trifft die Bedingung C_i für einen Aktuator zu, dann wird dieser so lange verzögert, bis der Aktuator zum Stillstand kommt. Dafür wird die Beschleunigung auf

$$\ddot{q}_{pos,i} = \begin{cases} -\ddot{q}_{brake,i} & \text{für } q_i > q_{m,i} \\ \ddot{q}_{brake,i} & \text{für } q_i \leq q_{m,i} \end{cases} \quad \text{mit } q_{m,i} = \frac{q_{max,i} - q_{min,i}}{2} \quad (4.4)$$

gesetzt. Das heißt, die Beschleunigung wird auf einen negativen Wert gestellt, wenn der Aktuator näher an der maximalen Position $q_{max,i}$ ist und auf einen positiven Wert, wenn sich der Aktuator näher an der minimalen Position $q_{min,i}$ befindet.

Um die Beschleunigung über die Geschwindigkeitsebene zu begrenzen kann

$$\dot{q}_i = \dot{q}_{i,old} + \ddot{q}_{pos,i} \Delta t \quad (4.5)$$

verwendet werden.

Geschwindigkeitsbeschränkung. Um die Geschwindigkeit zu beschränken, wird eine Skalierung durchgeführt [4]. Diese sorgt dafür, dass die Richtung des Geschwindigkeitsvektors beibehalten wird und sich nur der Betrag verkleinert. Folglich wird die Trajektorie mit einer niedrigeren Geschwindigkeit abgefahren. Durch die Koeffizienten

$$\gamma_i = \begin{cases} \frac{|\dot{q}_i|}{\dot{q}_{max,i}} & \text{für } |\dot{q}_i| > \dot{q}_{max,i} \\ 1 & \text{sonst} \end{cases} \quad (4.6)$$

lässt sich der Skalierungsfaktor $\gamma = \max_i(\gamma_i)$ bestimmen. Daraus kann mit

$$\dot{\mathbf{q}}_{vel} = \frac{1}{\gamma} \dot{\mathbf{q}}. \quad (4.7)$$

eine beschränkte Geschwindigkeit $\dot{\mathbf{q}}_{vel}$ berechnet werden, welche das Limit $\dot{q}_{max,i}$ nicht überschreitet. Es ist festzuhalten, dass mit der hier beschriebenen Geschwindigkeitsbeschränkung nur symmetrische Geschwindigkeitslimits vorgegeben werden können, also $\dot{q}_{max,i} = -\dot{q}_{min,i}$.

Beschleunigungsbeschränkung. Zur Beschränkung der Beschleunigung muss diese zuerst über den Differenzenquotienten

$$\ddot{q}_i = \frac{\dot{q}_i - \dot{q}_{i,old}}{\Delta t} \quad (4.8)$$

berechnet werden. Dabei bezeichnet \dot{q}_i die aktuelle Geschwindigkeit, $\dot{q}_{i,old}$ die Geschwindigkeit aus dem letzten Zeitschritt und Δt die Abtastzeit. Um die Beschleunigung auf Geschwindigkeitsebene zu begrenzen, kann folgende Gleichung verwendet werden [4]:

$$\dot{\mathbf{q}}_{acc} = \begin{cases} \dot{q}_{i,old} + \text{sign}(\ddot{q}_i) \ddot{q}_{max,i} \Delta t & \text{für } \ddot{q}_i > \ddot{q}_{max,i} \\ \dot{q}_i & \text{sonst} \end{cases}. \quad (4.9)$$

An dieser Stelle ist festzuhalten, dass sich durch die Beschränkung der Beschleunigung auch die Richtung der vorgegebenen Geschwindigkeit ändert. Außerdem muss bei einer kombinierten Beschränkung der Beschleunigung und Position, durch den oben erklärten Ansatz aufgepasst werden, dass die Bremsbeschleunigung $\ddot{q}_{brake,i}$ nicht größer ist wie die maximale Beschleunigung $\ddot{q}_{max,i}$, da ansonsten die Positionsbeschränkung nicht mehr eingehalten werden kann.

4.2. Definition der Aufgaben

Für die Bewegungsplattform werden in der Gelenkbahnplanung zwei Fälle unterschieden: Ob die Positionslimits der Aktuatoren des Hexapods eingehalten werden oder nicht. Je nachdem, welcher Fall vorliegt, werden in der NSB Steuerung verschiedene Aufgaben definiert.

Fall 1: Positionslimits werden eingehalten.

Werden alle Positionslimits eingehalten, müssen nur zwei Aufgaben definiert werden. Die erste Aufgabe σ_A mit der höheren Priorität sorgt dafür, dass die Bewegungsplattform einer Trajektorie folgt. Das bedeutet, dass diese mit der direkten Kinematik $\mathbf{w}_h = \mathbf{f}(\mathbf{q}_h)$ der Bewegungsplattform gleichgesetzt wird. Durch die zweite Aufgabe σ_o wird versucht, die Positionen der Aktuatoren des Hexapods im Nullraum von den Limits weg zu bewegen. Dadurch ergibt sich mit der NSB Steuerung aus (3.47) für die Gelenkgeschwindigkeit der Bewegungsplattform:

$$\begin{aligned}\dot{\mathbf{q}}_U &= \dot{\mathbf{q}}_A + \mathbf{N}_A \dot{\mathbf{q}}_o \\ &= \mathbf{J}_g^\dagger \dot{\mathbf{x}}_{A,mod} + \mathbf{N}_g \mathbf{W}^{-1} \dot{\mathbf{q}}_0,\end{aligned}\quad (4.10)$$

wobei $\dot{\mathbf{q}}_A$ die Geschwindigkeit und \mathbf{N}_A den Nullraumprojektor von Aufgabe σ_A darstellt.

Um die Geschwindigkeit von Aufgabe σ_A zu berechnen, wird nicht wie in der NSB Steuerung üblich $\dot{\mathbf{q}}_A = \mathbf{J}_A^\dagger \dot{\sigma}_A$ benutzt. Das hat den Grund, dass die Jacobi-Matrix \mathbf{J}_A nicht durch analytisches Ableiten von σ_A nach den Minimalkoordinaten \mathbf{q}_h gebildet werden kann, weil die Bewegungsplattform eine hybride Struktur besitzt. Stattdessen wird die geometrische Jacobi-Matrix \mathbf{J}_g der Bewegungsplattform verwendet, deren Berechnung bereits in Abschnitt 3.4.2 erklärt wurde. Um die Geschwindigkeit $\dot{\mathbf{q}}_A$ zu erhalten, wird diese mit der modifizierten Geschwindigkeit $\dot{\mathbf{x}}_{A,mod}$ aus (3.54) multipliziert. Die dafür benötigte Winkelgeschwindigkeit ω_A kann durch (3.55) berechnet werden.

Es besteht die Möglichkeit, den letzten Geschwindigkeitsvektor der NSB Steuerung frei zu wählen, weshalb die Geschwindigkeit $\dot{\mathbf{q}}_o$ durch die Nullraumgeschwindigkeit $\dot{\mathbf{q}}_0$ aus (3.43) ersetzt wird. Diese sorgt durch den Nullraumprojektor \mathbf{N}_g dafür, dass die Positionen der Aktuatoren im Nullraum von den Limits weg gedrückt werden, was genau den Zweck von Aufgabe σ_o erfüllt. Diese wird zusätzlich durch die Matrix \mathbf{W}^{-1} gewichtet, wodurch sich das „Automatic Supervisory Control“ aus (3.42) ergibt.

Fall 2: Positionslimits werden verletzt.

Existiert eine Anzahl u an Aktuatoren des Hexapods, welche gesperrt werden müssen, um die Positionslimits einzuhalten, so muss eine Aufgabe $\sigma_l \in \mathbb{R}^u$ als „set-based task“ definiert werden. Diese wird mit den Minimalkoordinaten q_i der gesperrten Aktuatoren gleichgesetzt. Zum besseren Verständnis soll hier beispielhaft der Fall betrachtet werden, bei dem die Aktuatoren eins, drei und fünf gesperrt sind. Dafür ergibt sich die Aufgabe

$$\sigma_l = \begin{pmatrix} q_{h,1} \\ q_{h,3} \\ q_{h,5} \end{pmatrix}, \quad (4.11)$$

wovon die Jacobi-Matrix $\mathbf{J}_l \in \mathbb{R}^{u \times n}$ gebildet wird:

$$\mathbf{J}_l = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \quad (4.12)$$

Wie in der Matrix \mathbf{J}_l zu erkennen ist, ergibt sich als Jacobi Matrix für jeden Aktuator, der in σ_l eingetragen ist, ein transponierter Einheitsvektor $\mathbf{e} \in \mathbb{R}^n$.

Um die Aktuatoren so spät wie möglich zu bremsen, wird die Bang-Bang Steuerung eingesetzt, welche bereits in Abschnitt 4.1 beschrieben wurde. Der daraus berechnete Bremsweg $s_{brake,i}$ (siehe (4.2)) dient zur Aktivierung des „set-based task“ und wird für den ε -Wert aus Abbildung 10 verwendet.

Um die Geschwindigkeit $\dot{\mathbf{q}}_l = \mathbf{J}_l \dot{\sigma}_l$ für die NSB Steuerung zu berechnen, wird die Ableitung $\dot{\sigma}_l$ auf die in (4.5) berechnete Geschwindigkeit für jeden Aktuator gesetzt. Dadurch werden diese mit der Bremsbeschleunigung \ddot{q}_{brake} abgebremst. Befindet sich ein Aktuator am Limit im Stillstand, wird die Ableitung $\dot{\sigma}_l$ auf null gesetzt und auf einen Befehl gewartet, der den Aktuator vom Limit weg drückt. Sobald der Befehl gegeben wird, wird der Aktuator aus der Aufgabe σ_l gelöscht.

Wie bereits erwähnt, ist es in der Bewegungssimulation wichtiger, die Orientierung einzuhalten, als die Position, weshalb zwei Aufgaben für die Trajektorie definiert werden. Die erste Aufgabe σ_B wird zur Einhaltung der Orientierung und die zweite Aufgabe σ_C zur Einhaltung der Position verwendet. Es ergeben sich dadurch im Gegensatz zu (4.10) keine großen Veränderungen, es muss nur die geometrische Jacobi-Matrix \mathbf{J}_g in die Jacobi-Matrix der Rotation $\mathbf{J}_{R,\omega}$ und die Jacobi-Matrix der Translation \mathbf{J}_T aufgeteilt werden. Für die modifizierte Geschwindigkeit $\dot{\mathbf{x}}_{A,mod}$ erfolgt die gleiche Aufteilung. Es wird außerdem noch die Aufgabe σ_o definiert, welche die gleiche Funktion besitzt wie für den Fall ohne beschränkte Aktuatoren.

Durch einsetzen der definierten Aufgaben in die NSB Steuerung ergibt sich für die Geschwindigkeit der Minimalkoordinaten:

$$\begin{aligned} \dot{\mathbf{q}}_L &= \dot{\mathbf{q}}_l + \mathbf{N}_{l,l} \dot{\mathbf{q}}_B + \mathbf{N}_{l,B} \dot{\mathbf{q}}_C + \mathbf{N}_{l,C} \dot{\mathbf{q}}_o \\ &= \mathbf{J}_l^\dagger \dot{\sigma}_l + \mathbf{N}_{l,l} \mathbf{J}_{R,\omega}^\dagger \dot{\mathbf{r}}_{B,mod} + \mathbf{N}_{l,B} \mathbf{J}_T^\dagger \boldsymbol{\omega}_{C,mod} + \mathbf{N}_{l,C} \dot{\mathbf{q}}_o, \end{aligned} \quad (4.13)$$

mit:

$$\mathbf{J}_{l,C} = \begin{pmatrix} \mathbf{J}_l \\ \mathbf{J}_B \\ \mathbf{J}_C \end{pmatrix}. \quad (4.14)$$

Für den zweiten Fall wird die Gewichtungsmatrix \mathbf{W} auf die Einheitsmatrix gesetzt. Außerdem wird die Pseudoinverse mithilfe der Singulärwertzerlegung (engl. Singular Value Decomposition) berechnet [52].

4.3. Funktionsweise der Gelenkbahnplanung

In diesem Abschnitt wird die Funktionsweise der Gelenkbahnplanung anhand der zwei Flussdiagramme aus Abbildung 13 erklärt.

Im ersten Flussdiagramm in Abbildung 13a ist der Ablauf der Gelenkbahnplanung für einen Zeitschritt dargestellt. Als Eingang dient, wie schon in Abbildung 12 gezeigt, die gewünschte Position

$\mathbf{r}_d(t)$ und die gewünschte Orientierung $\mathbf{A}_d(t)$, sowie die gemessenen Minimalkoordinaten \mathbf{q}_{meas} der Bewegungsplattform. Daraus wird durch (4.10) die Minimalgeschwindigkeit $\dot{\mathbf{q}}_U$, für den Fall, dass keine Aktuatoren des Hexapods beschränkt sind, berechnet.

In der Parametrierung, welche am Ende dieses Abschnitts erklärt wird, kann festgelegt werden, ob die Positionen \mathbf{q} , die Geschwindigkeiten $\dot{\mathbf{q}}$ oder die Beschleunigungen $\ddot{\mathbf{q}}$ beschränkt werden sollen. Bei aktivierter Positionsbeschränkung wird das Unterprogramm „Position beschränken“ ausgeführt, das im Flussdiagramm in Abbildung 13b dargestellt ist. Weiter wird die Geschwindigkeit $\dot{\mathbf{q}}$ durch (4.7) und die Beschleunigung $\ddot{\mathbf{q}}$ durch (4.9) begrenzt, wenn dies in der Parametrierung angegeben ist.

Da die Positions-, Geschwindigkeits- und Beschleunigungsbeschränkungen zeitdiskret berechnet werden, wird die resultierende Minimalgeschwindigkeit $\dot{\mathbf{q}}$ gefiltert, um den Verlauf zu glätten und anschließend integriert. Es resultieren die Minimalkoordinaten \mathbf{q} , welche an den Steuercomputer übergeben werden.

Im Folgenden wird das Unterprogramm „Position beschränken“, welches im Flussdiagramm in Abbildung 13b dargestellt ist, genauer beschrieben.

Zuerst wird im Algorithmus 1 durch die Position \mathbf{q} und die Geschwindigkeit $\dot{\mathbf{q}}_U$ überprüft, ob Positionslimits des Hexapods verletzt werden. Befindet sich ein Aktuator an einem Positionslimit und wird durch $\dot{\mathbf{q}}_U$ eine Geschwindigkeit vorgegeben, welche diesen vom Limit zurück in den erlaubten Bereich bewegt, wird der Aktuator nicht beschränkt bzw. die Beschränkung aufgehoben. Das Ergebnis, ob ein Aktuator des Hexapods beschränkt werden muss, wird in $\mathbf{PL1} \in \mathbb{R}^6$ abgespeichert. Zusätzlich muss überprüft werden, welche Aktuatoren im letzten Zeitschritt abgebremst wurden, aber noch nicht vollständig zum Stillstand gekommen sind. Diese Aktuatoren werden in $\mathbf{PL1}$ ebenfalls als „zu beschränken“ festgelegt. Diese Überprüfung ist für die Positionsbeschränkung durch die Bang-Bang Steuerung, die in Abschnitt 4.1 erklärt wurde, wichtig, da ein Aktuator der einmal zu bremsen begonnen hat vollständig abgebremst werden muss. Es besteht sonst die Möglichkeit, dass das Positionslimit verletzt wird.

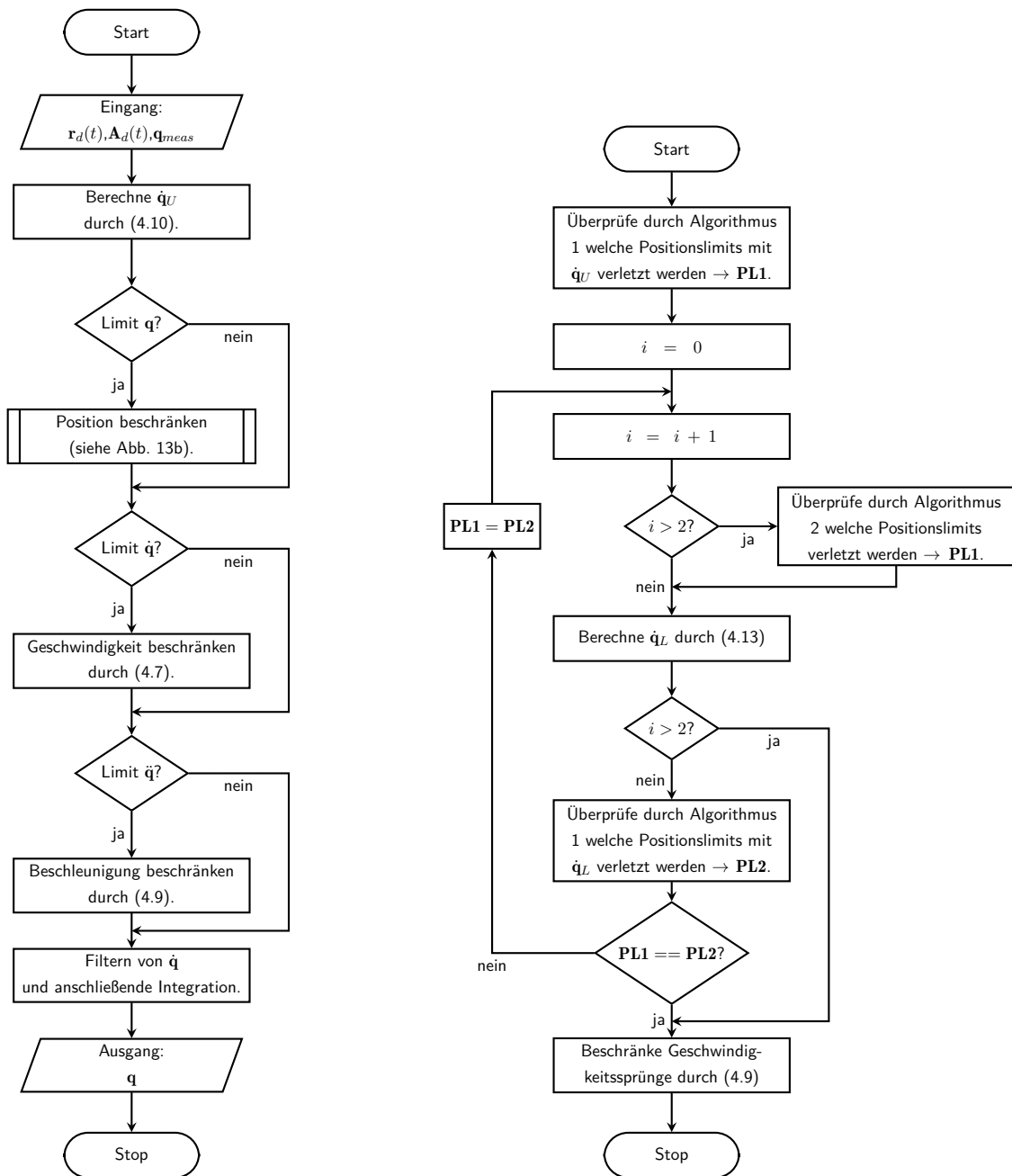
Als nächstes wird die Geschwindigkeit $\dot{\mathbf{q}}_L$ durch (4.13) berechnet. Die Aufgabe σ_l zur Beschränkung der Aktuatoren wird mithilfe von $\mathbf{PL1}$ definiert.

Im Anschluss wird mit der neu berechneten Geschwindigkeit $\dot{\mathbf{q}}_L$ durch Algorithmus 1 erneut überprüft, welche Aktuatoren gesperrt werden müssen. Das Ergebnis wird in Vektor $\mathbf{PL2} \in \mathbb{R}^6$ abgespeichert. Auch hier werden die Aktuatoren, die im letzten Zeitschritt gebremst wurden aber nicht zum Stillstand gekommen sind, hinzugefügt. Die erneute Überprüfung muss durchgeführt werden, da sich durch das Verändern der Aufgabe σ_l ein Vorzeichen in $\dot{\mathbf{q}}_L$ ändern kann, wodurch die Möglichkeit besteht, dass ein Positionslimit verletzt wird, das durch die zuvor berechnete Geschwindigkeit $\dot{\mathbf{q}}_U$ nicht verletzt wurde.

Stimmen die beiden Vektoren $\mathbf{PL1}$ und $\mathbf{PL2}$ überein, werden etwaige Geschwindigkeitssprünge, welche durch Veränderung der Aufgabe σ_l in Gleichung (4.13) auftreten können, durch die Beschleunigungsbeschränkung aus (4.9) begrenzt. Anschließend wird die neu berechnete Geschwindigkeit ausgegeben.

Stimmen die Vektoren $\mathbf{PL1}$ und $\mathbf{PL2}$ nicht überein, wird $\mathbf{PL1} = \mathbf{PL2}$ gesetzt und i um eins hochgezählt. Dann wird die Geschwindigkeit $\dot{\mathbf{q}}_L$ erneut berechnet und wieder überprüft, ob die beiden

Vektoren identisch sind. Falls nach drei Durchläufen ($i > 2$) immer noch keine Übereinstimmung vorliegt, wird **PL1** mithilfe von Algorithmus 2 berechnet. Dabei wird nur die Position als Kriterium verwendet um die Aktuatorbeschränkungen festzulegen, wodurch eine Einhaltung der Positionslimits garantiert ist. Es ist festzuhalten, dass es nicht möglich ist, immer Algorithmus 2 anstelle von Algorithmus 1 zu verwenden. Das würde bedeuten, dass ein Aktuator, welcher sich an einem Positionslimit befindet dauerhaft dort festgehalten werden würde, da dieser durch Überprüfung mit Algorithmus 2 immer gesperrt bleibt.



(a) Flussdiagramm für die Berechnung der Minimalkoordinaten q in einem Zeitschritt.

(b) Flussdiagramm für das Unterprogramm „Position beschränken“ aus Abbildung 13a.

Abbildung 13 Flussdiagramme für Gelenkbahnplanungsmodul.

Algorithmus 1 Ein Algorithmus, mit dem unter Berücksichtigung der Geschwindigkeit überprüft wird, ob Aktuatoren beschränkt werden müssen.

Eingang: $q, \dot{q}, q_{min}, q_{max}, \varepsilon, s$

$$q_{a,min} = q_{min} + \varepsilon + s$$

$$q_{a,max} = q_{max} - \varepsilon - s$$

```
if ( $q \leq q_{a,min}$  and  $\dot{q} < 10^{-8}$ ) or ( $q \geq q_{a,max}$  and  $\dot{q} > -10^{-8}$ ) then
    return True
else
    return False
end if
```

Algorithmus 2 Ein Algorithmus, mit dem ohne Berücksichtigung der Geschwindigkeit überprüft wird, ob Aktuatoren beschränkt werden müssen.

Eingang: $q, q_{min}, q_{max}, \varepsilon, s$

$$q_{a,min} = q_{min} + \varepsilon + s$$

$$q_{a,max} = q_{max} - \varepsilon - s$$

```
if  $q_{a,min} < q$  and  $q < q_{a,max}$  then
    return False
else
    return True
end if
```

5. Implementierung

In diesem Kapitel wird in Abschnitt 5.1 die Implementierung eines Mehrkörpersimulationsmodells der 8-Achsen Bewegungsplattform vorgestellt, wofür anschließend in Abschnitt 5.2 eine Visualisierung präsentiert wird. Zum Schluss wird in Abschnitt 5.3 auf das implementierte Gelenkbahnplanungsmo-
dul eingegangen und das Simulationsmodell vorgestellt, mit dem dieses in Kapitel 6 verifiziert wird. Das Mehrkörpersimulationsmodell wird in Kapitel 7 außerdem für die Auslegung der Achsen 7 und 8 verwendet.

Für die Implementierung wurde im Zuge dieser Masterarbeit die Modellierungs- und Simulationsumgebung Dymola¹ von Dassault Systems verwendet, die Modelica verwendet. Bei Modelica handelt es sich um eine offene, objektorientierte Programmiersprache, welche zum Modellieren von dynamischen Systemen eingesetzt wird [19].

5.1. Mehrkörpersimulationsmodell

In diesem Abschnitt wird der Aufbau des Mehrkörpersimulationsmodells der 8-Achs Bewegungsplattform erklärt. Dieses wurde mithilfe der Modelica-MultiBody-Bibliothek erstellt, die einen Teil der Modelica-Standard-Bibliothek darstellt. Damit ist es möglich, dreidimensionale mechanische Systeme zu modellieren und zu simulieren [43].

Bevor auf das Mehrkörpersimulationsmodell der gesamten 8-Achs Bewegungsplattform eingegangen wird, wird zunächst das in Abbildung 14 dargestellte Simulationsmodell des Hexapods näher erläutert.

Die beiden Blöcke „basePlattform“ und „payloadPlattform“ stellen die obere und die untere Plattform des Hexapods dar. Beide Plattformen besitzen jeweils sechs Befestigungspunkte für die Aktuatoren, welche im Modell durch die Blöcke „cylinder1“ bis „cylinder6“ dargestellt sind.

Die „cylinder“-Blöcke sind nach dem Vorbild der Aktuatoren aus Abbildung 2 aufgebaut. Sie bestehen deshalb aus einem prismatischen Gelenk für das eine Position vorgegeben werden kann, sowie aus zwei Universalgelenken an den beiden Enden des prismatischen Gelenks. Die Universalgelenke sind mit den Ausgängen des „cylinder“-Blocks verbunden, welche wiederum mit den Befestigungspunkten der beiden Plattformen verbunden sind. Ein Aktuator weist daher mit den zwei Freiheitsgraden jedes Universalgelenks und dem des prismatischen Gelenks fünf Freiheitsgrade auf. Mit der Grüblerschen Gleichung kann gezeigt werden, dass jeder Aktuator einen weiteren Freiheitsgrad benötigt, damit das gesamte System insgesamt sechs Freiheitsgrade besitzt. Am realen Hexapod kann bei sich bewegenden Aktuatoren eine Verdrehung der Kolbenstange zum Zylinderrohr beobachtet werden, weshalb im Modell ein Drehgelenk zwischen das prismatische Gelenk und das obere Universalgelenk eingesetzt wird. An den Befestigungspunkten der oberen Plattform entsteht dadurch jeweils ein Kugelgelenk.

¹ <https://www.3ds.com/products-services/catia/products/dymola/>

Durch den Eingang „ $q[]$ “ können die Aktuatorpositionen vorgegeben werden und durch den Ausgang „ $q_measured[]$ “ die tatsächlichen Aktuatorpositionen ausgelesen werden. Im Moment wird ein idealisiertes Modell angenommen, bei dem die Aktuatoren direkt der vorgegebenen Position folgen, weshalb „ $q[]$ “ und „ $q_measured[]$ “ immer identisch sind. Der Ausgang „ $q_measured[]$ “ unterscheidet sich erst von „ $q[]$ “, wenn in zukünftigen Erweiterungen des Modells zum Beispiel ein Regler für die Positionsvorgabe der Aktuatoren eingesetzt wird. Sogenannte „Frames“ dienen in der Modelica-MultiBody-Bibliothek als Schnittstelle, um Körper und Gelenke miteinander zu verbinden. Damit werden die Position und die Orientierung, sowie die Schnittkräfte und Schnittmomente ausgetauscht. Mithilfe von „ $frame_base$ “ kann die untere Plattform des Hexapods mit einem anderen Körper oder einem Inertialsystem verbunden werden. Weiter ist es durch „ $frame_payload$ “ möglich, Körper auf dem Hexapod zu befestigen, sowie zum Beispiel die Position und Orientierung der oberen Plattform auszulesen. Um den Hexapod mithilfe der in Abschnitt 5.2 gezeigten Visualisierung darstellen zu können, werden „ $frame_cylinderStarts[]$ “ und „ $frame_cylinderEnds[]$ “ verwendet. Durch den Block „ $PayloadPlatform$ “ wird die Masse und der Trägheitstensor der oberen Plattform des Hexapods festgelegt. Auch in den „ $cylinder$ “-Blöcken ist die Masse und der Trägheitstensor für das Zylinderrohr und die Kolbenstange definiert. Da der Hexapod mit dem Boden verschraubt ist, wurden für die untere Plattform keine Massendaten festgelegt. Für die Bestimmung der Massen und der Trägheitstensoren wurde das CAD-Modell des Hexapods verwendet.

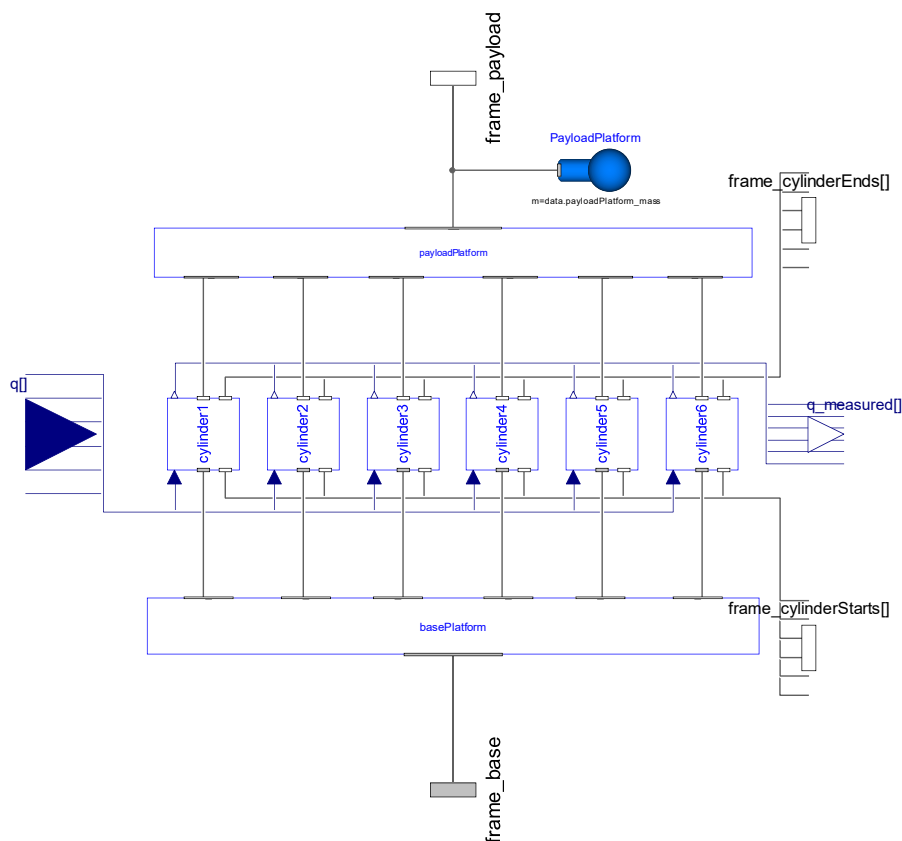


Abbildung 14 Aufbau des Hexapod Mehrkörpersimulationsmodells in Modelica.

Das Mehrkörpersimulationsmodell der gesamten 8-Achs Bewegungsplattform ist in Abbildung 15 dargestellt. Darin sind auf dem Modell des Hexapods aus Abbildung 14 zwei weiteren Drehachsen („axis7“ und „axis8“) befestigt, deren Achswinkel vorgegeben werden können. Die Position und Orientierung dieser Drehachsen kann jeweils durch eine feste Verschiebung und Verdrehung definiert werden, welche so gewählt wurden, wie in Abbildung 6 gezeigt. Mithilfe des „fixedRotation“-Blocks kann die relative Position und Orientierung des Endeffektors zu Achse 8 definiert werden. Weiter werden durch die drei Blöcke „PayloadFrame“, „Axis7“ und „Axis8“ die Massen und Trägheitstensenoren für die Körper der 8-Achs Bewegungsplattform vorgegeben, welche mithilfe eines CAD-Modells ermittelt wurden.

Auch im Gesamtmodell können wie im Modell des Hexapods die Aktuatorpositionen durch den Eingang „q[]“ vorgegeben werden und die tatsächlichen Aktuatorpositionen durch den Ausgang „q_measured[]“ ausgelesen werden. Um die Basis des Simulators festzulegen, wird „frame_base“ benutzt, der Endeffektor wird durch „frame_TCP“ beschrieben. Die fünf „Frames“ auf der rechten Seite von Abbildung 15 werden für die Visualisierung benötigt, die in Abschnitt 5.2 näher beschrieben wird.

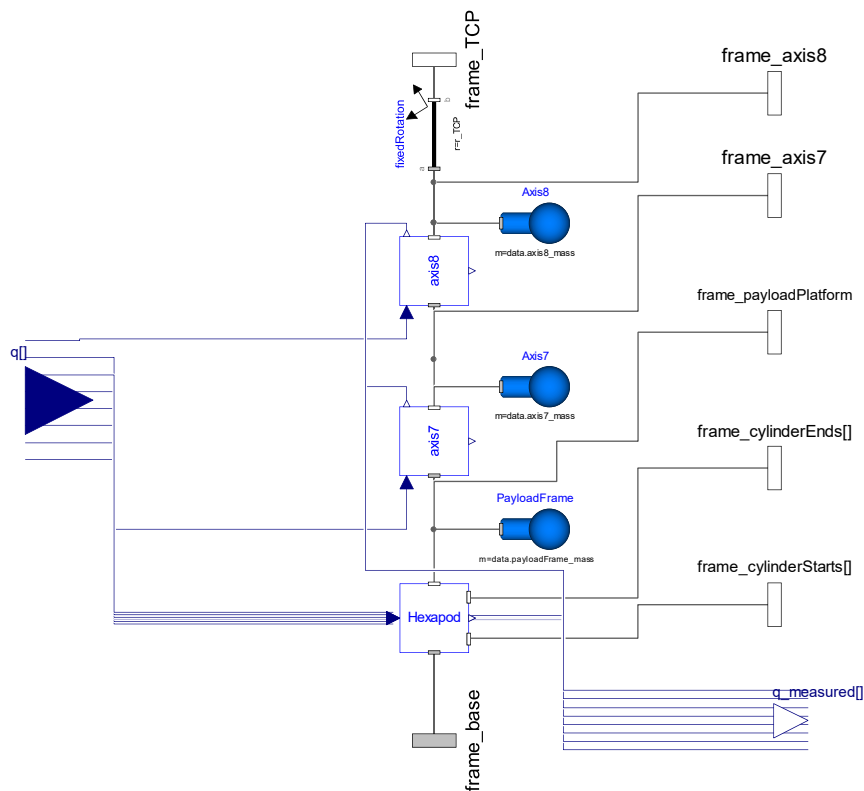


Abbildung 15 Aufbau des Mehrkörpersimulationsmodells der 8-Achs Bewegungsplattform in Modelica.

Für die Simulation des Modells ist es wichtig, Startwerte für die Drehgelenke und Universalgelenke der „cylinder“-Blöcke aus Abbildung 14 vorzugeben. Diese müssen möglichst nahe an der gewünschten Startkonfiguration des Systems liegen, um zu vermeiden, dass die Aktuatoren des Hexapods in einer ungeeigneten Konfiguration starten.

Weiter ist festzuhalten, dass es im Rahmen dieser Masterarbeit nicht möglich war, eine Validierung des Mehrkörpersimulationsmodells des Hexapods am realen System durchzuführen, da dieser noch nicht betriebsbereit ist. Die Validierung sollte einer der ersten Schritte sein, sobald der Hexapod im Betrieb ist.

5.2. Visualisierung

Um die Simulationsergebnisse der 8-Achs Bewegungsplattform darstellen zu können, wurde mithilfe der DLR Visualization-Bibliothek² eine Visualisierung erstellt, die in Abbildung 16 abgebildet ist. Dafür wird von jedem Körper eine STL-Datei benötigt, welche die Geometrie beschreibt. Diese Datei wird aus dem CAD-Modell erstellt, wobei darauf geachtet werden sollte bezüglich welchem Koordinatensystem die STL-Datei definiert wird, um die Positionierung in der Visualisierung zu vereinfachen.

Für die Übergabe der Bewegungen der Körper an die Visualisierung werden die fünf „Frames“ des Simulationsmodells der Bewegungsplattform benutzt, die in Abbildung 15 auf der rechten Seite dargestellt sind. In Abbildung 17 ist ein Beispiel abgebildet, welches zeigt, wie die Visualisierung in dieser Arbeit eingesetzt wird.

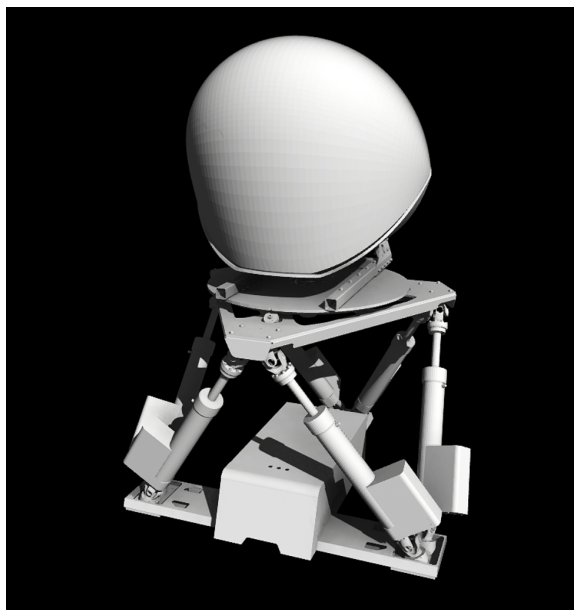


Abbildung 16 Screenshot der Visualisierung der 8-Achs Bewegungsplattform.

5.3. Gelenkbahnplanungsmodul

Die Funktionsweise der Gelenkbahnplanung wurde bereits in Abschnitt 4.3 erklärt. In diesem Abschnitt wird das Simulationsmodell aus Abbildung 17 beschrieben, welches im nächsten Kapitel verwendet wird, um das Gelenkbahnplanungsmodul zu verifizieren. Im Anschluss wird auf die Para-

² <https://www.systemcontrolinnovationlab.de/the-dlr-visualization-library/>

metrierung des Gelenkbahnplanungsmoduls eingegangen.

In Abbildung 17 ist ein Simulationsmodell abgebildet, mit dem eine kartesische Trajektorie auf dem Mehrkörpersimulationsmodell abgefahren werden kann. Durch den Block „combiTimeTable“ wird die kartesische Trajektorie in die Simulation eingelesen. Diese besteht aus 12 Werten, von denen drei die gewünschte Position beschreiben. Die restlichen neun Werte beschreiben die Einträge einer Rotationsmatrix, welche die gewünschte Orientierung abbildet. Durch den „Trajectory“-Block werden die 12 Werte der Trajektorie in einen Vektor für die Position und eine Matrix für die Orientierung umgewandelt. Anschließend werden diese an das Gelenkbahnplanungsmodul ($q=f(r,T)$) weitergegeben, das aus der kartesischen Trajektorie die Aktuatorpositionen für die Bewegungsplattform ermittelt. Es besteht die Möglichkeit dieses Gelenkbahnplanungsmoduls zu parametrieren, worauf am Ende dieses Kapitels näher eingegangen wird. Da im Gelenkbahnplanungsmodul eine inverse Kinematik mit Rückführung verwendet wird (siehe Abschnitt 3.6), werden aus dem Mehrkörpersimulationsmodell (Simulator) die gemessenen Aktuatorpositionen zurückgeführt. Außerdem wird das Simulationsmodell durch den „Visualization“-Block in der Visualisierung dargestellt, welche durch den „visualizationSetup“-Block parametrieren werden kann. Der „world“-Block stellt ein Inertialsystem dar und beschreibt die Umgebung.

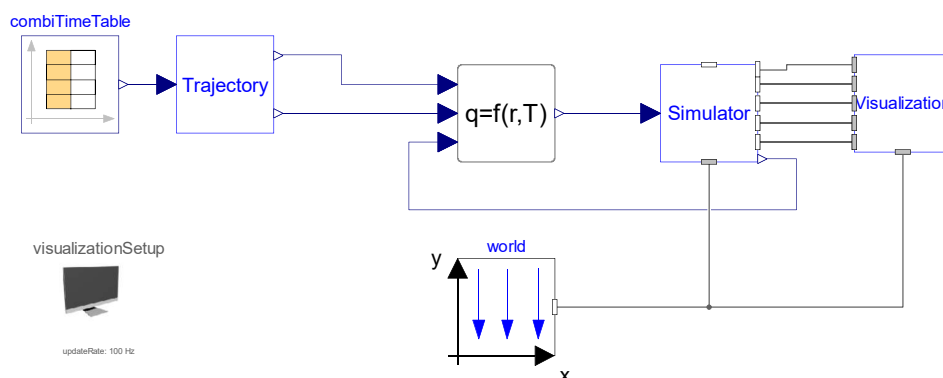


Abbildung 17 Aufbau der Simulation einer kartesischen Trajektorie mit dem Gelenkbahnplanungsmodul in Modelica.

Parameter

Die verwendeten Parameter für das implementierte Gelenkbahnplanungsmodul sind in Abbildung 18 dargestellt und sind in fünf Gruppen aufgeteilt: Allgemein, Initialisierung, Gewichtung, Beschränkung und Filter.

In den allgemeinen Parametern kann durch „r_TCP“ die Position des Endeffektors relativ zum Flansch von Achse 8 vorgegeben werden. Außerdem wird durch „sampleTime“ die Abtastzeit festgelegt, welche für die zeitdiskrete Berechnung der Aktuatorbeschränkungen genutzt wird.

In der Gruppe Initialisierung werden mit „q_start“ die Startwerte aller Aktuatoren für die Simulation vorgegeben, sowie die Startwerte für die Position ($r_guess_hexapod$) und die Orientierung ($angles_guess_hexapod$), die zur iterativen Lösung der direkten Kinematik des Hexapods benötigt werden. Die Orientierung wird durch Kardan-Winkel in der Reihenfolge x-Achse, y-Achse, z-Achse

angegeben.

Mithilfe der Gewichtungparameter wird die Berechnung der inversen Kinematik beeinflusst. Dort gibt „W“ die Einträge für die Diagonalmatrix \mathbf{W} zur Berechnung der gewichteten Moore-Penrose Pseudoinversen aus (3.41) an. Durch „K“ können die Diagonaleinträge der Gewichtungsmatrizen \mathbf{K}_r und \mathbf{K}_ω festgelegt werden, welche in der geschlossenen inversen Kinematik verwendet werden (siehe (3.54)). Die ersten drei Einträge geben die Gewichtung für die Position und die letzten drei die Gewichtung für die Orientierung an. Außerdem kann durch „alpha_pos“ und „alpha_vel“ die Nullraumgeschwindigkeit $\dot{\mathbf{q}}_0$ (siehe (3.43)) skaliert und durch „eps_pos“ und „eps_vel“ der Grenzwert definiert werden, ab dem die Nullraumgeschwindigkeit quadratisch ansteigt.

In der Beschränkungsgruppe werden verschiedene Grenzwerte für die Aktuatoren der Bewegungsplattform angegeben. Dabei kann durch „limitPosition“, „limitVelocity“ und „limitAcceleration“ eingestellt werden, ob eine Beschränkung der Position der Geschwindigkeit oder der Beschleunigung durchgeführt werden soll. Mit „Hexapod_actuator_max“ und „Hexapod_actuator_min“ werden die Maximalwerte der Aktuatoren des Hexapods festgelegt und durch „Hexapod_actuator_s“ wird eine Sicherheitszone als prozentualer Anteil des gesamten Hubs eines Aktuators angegeben. Diese Sicherheitszone ist in Abbildung 10 mit s beschrieben. Weiter wird mit „q_ddot_break“ die Beschleunigung angegeben mit der ein Aktuator bei einem Verstoß gegen das Positionslimit abgebremst wird (siehe (4.2)) und mit „q_ddot_change“ die maximale Beschleunigung, welche bei einem Geschwindigkeitssprung durch einen Aufgabenwechsel in (4.13) entstehen kann. Außerdem können durch „q_dot_max“ und „q_ddot_max“ die maximalen Geschwindigkeiten und Beschleunigungen der Aktuatoren festgelegt werden.

Zuletzt kann in der Filtergruppe eingestellt werden, ob die berechneten Gelenkgeschwindigkeiten $\dot{\mathbf{q}}$, vor der Integration gefiltert werden sollen. Es besteht die Möglichkeit der Verwendung eines Butterworth-Filters und eines Filters mit kritischer Dämpfung [49]. Wird ein Filter aktiviert, kann auch die Eckfrequenz (f) festgelegt werden. Es ist festzuhalten, dass durch die Filterung die Beschleunigungslimits verletzt werden können. Durch eine Reduktion der Beschleunigung um 10 Prozent hat sich gezeigt, dass die Beschleunigungslimits eingehalten werden.

5 Implementierung

Parameters		
r_TCP	<input type="text" value="data.rCT"/>	Translation vector from axis 8 frame to TCP frame.
sampleTime	<input type="text" value="0.004"/>	Sample time for time-discrete calculation of position, velocity and acceleration limits.
Initialization		
q_start	<input type="text" value="data.q_neutral"/>	Initial value for actuators.
r_guess_hexapod	<input type="text" value="data.r_neutral_hexa"/>	Initial position for direct kinematics of hexapod.
angles_guess_hexapod	<input type="text" value="data.angles_neutral_hexa"/>	Initial orientation for direct kinematics of hexapod (sequence={1,2,3}).
Weighting		
W	<input type="text" value="{1,1,1,1,1,1/5,1/5}"/>	Weights for each actuator. Lower value -> higher weighting.
K	<input type="text" value="10*{1,1,1,1,1}"/>	Gain vector for workspace error [x,y,z,omegax,omegay,omegaz].
alpha_pos	<input type="text" value="1"/>	Scaling factor for position part of nullspace velocity.
eps_pos	<input type="text" value="0.1"/>	Offset from limit where position limitation starts.
alpha_vel	<input type="text" value="0.5"/>	Scaling factor for velocity part of nullspace velocity.
eps_vel	<input type="text" value="{0.15,0.15,0.15,0.15,0.15,0.15,0.5,0.5}"/>	Offset from limit where velocity limitation starts.
Limits		
limitPosition	<input checked="" type="checkbox"/>	Activate or deactivate position limitation.
Hexapod_actuator_max	<input type="text" value="0.635"/>	Maximum stroke length of hexapod actuator.
Hexapod_actuator_min	<input type="text" value="0.02"/>	Minimum stroke length of hexapod actuator.
Hexapod_actuator_s	<input type="text" value="5"/>	Safety distance from hexapod limit. Percentage of total stroke.
q_ddot_break	<input type="text" value="{3.5,3.5,3.5,3.5,3.5,3.5}"/>	Acceleration limits for braking of hexapod actuators when reaching a position limit.
q_ddot_change	<input type="text" value="{3.5,3.5,3.5,3.5,3.5,2.1}"/>	Acceleration limits when differential equations are not continuous.
limitVelocity	<input checked="" type="checkbox"/>	Activate or deactivate velocity limitation.
q_dot_max	<input type="text" value="{0.45,0.45,0.45,0.45,0.45,0.45,4.01,2.26}"/>	Velocity limit for each actuator.
limitAcceleration	<input checked="" type="checkbox"/>	Activate or deactivate acceleration limitation.
q_ddot_max	<input type="text" value="0.9*{7.76,7.76,7.76,7.76,7.76,7.76,1.75,2.26}"/>	Acceleration limit for each actuator.
Filter		
useFilter	<input type="radio"/> No Filter <input type="radio"/> lowpassButterworth <input checked="" type="radio"/> criticalDamping	Type of Filter to use.
f	<input type="text" value="20"/>	Cut-off frequency if filter is selected.

Abbildung 18 Screenshot der Parameter des Gelenkbahnplanungsmoduls aus Dymola.

6. Verifizierung der Gelenkbahnplanung

Mithilfe des Mehrkörpersimulationsmodells der 8-Achs Bewegungsplattform (siehe Abschnitt 5.1) wird in diesem Kapitel die implementierte Gelenkbahnplanung (siehe Kapitel 4) verifiziert. Dafür wird das in Abschnitt 5.3 vorgestellte Modelica-Simulationsmodell verwendet, mit dem eine gewünschte kartesische Trajektorie für den Endeffektor der Bewegungsplattform vorgegeben werden kann.

Im Folgenden wird in Abschnitt 6.1 die vorgegebene Trajektorie erläutert und in Abschnitt 6.2 die verwendete Parametrierung vorgestellt. Abschließend wird in Abschnitt 6.3 die Gelenkbahnplanung validiert und analysiert.

6.1. Erklärung der vorgegebenen Trajektorie

Als kartesische Trajektorie wird ein sogenannter VDA-Spurwechseltest (früher Elchtest genannt) vorgegeben, welcher in der ISO 3888-2 definiert ist. Dabei handelt es sich um ein in der Automobilindustrie übliches Fahrmanöver zum Testen der Fahrstabilität von PKWs. Für das Fahrmanöver wird ungebremst und unter hoher Geschwindigkeit zuerst ein Fahrspurwechsel nach links und nach einer kurzen Geradeausfahrt ein Spurwechsel nach rechts durchgeführt. Um den VDA-Spurwechseltest im Arbeitsraum der 8-Achs Bewegungsplattform abbilden zu können, wurden die Messdaten des Tests durch den klassischen Washoutfilter (CWOFF) nach Reid und Nahon [48] angepasst.

In Abschnitt 6.3 werden zwei durch den Washoutfilter unterschiedlich parametrisierte Trajektorien des VDA-Spurwechseltests vorgestellt. Mit der ersten Trajektorie (Parametrierung nach [48] Tabelle 5.3, Set F), welche sich vollständig im Arbeitsraum der Bewegungsplattform darstellen lässt, wird demonstriert, dass die Gelenkbahnplanung die Bewegungen richtig abbildet. Anhand der zweiten Trajektorie (Parametrierung nach [48] Tabelle 5.3, Set A), welche die Arbeitsraumgrenzen der Bewegungsplattform verletzt, wird das Verhalten der Gelenkbahnplanung bei Verstößen gegen die implementierten Aktuatorbeschränkungen gezeigt. Im weiteren Verlauf dieses Kapitels wird die Anwendung der ersten Trajektorie auf das Simulationsmodell aus Abbildung 17 als Fall A und die Anwendung der zweiten Trajektorie als Fall B bezeichnet.

6.2. Parametrierung der Gelenkbahnplanung

Die Parameter der Gelenkbahnplanung wurden für Fall A und Fall B wie in Abbildung 18 dargestellt gewählt.

Dort ist die „sampleTime“ auf 0.004s gesetzt, was einer Frequenz von 250Hz entspricht. Das ist gleichzeitig die maximale Frequenz für den Datenaustausch mit der Steuerung des Hexapods. Die Initialisierungsparameter wurden entsprechend der Neutralstellung des Hexapods gewählt, wobei sich alle Aktuatoren in der mittleren Position befinden. Dabei handelt es sich um die Position, die der Hexapod nach dem Einschalten der Motorsteuerung einnimmt.

Um das Verhalten der Gelenkbahnplanung zu beeinflussen, sind primär die Gewichtungparameter (Weighting) und die Beschränkungsparameter (Limits) verantwortlich. Die Beschränkungsparameter

parameter wurden zum größten Teil aus den Maschinendaten des Hexapods übernommen. Für die Geschwindigkeits- und Beschleunigungslimits von Achse 7 und 8 wurden die in Kapitel 7 definierten Werte verwendet.

Die Gewichtungparameter wurden durch Versuch und Irrtum bestimmt, deren Einfluss im Folgenden näher beschrieben wird.

Wie in Abschnitt 5.3 schon dargelegt, können durch den Parameter „W“ die einzelnen Aktuatoren der Bewegungsplattform gewichtet werden. Diese Gewichtung kann vorteilhaft eingesetzt werden, wenn eine Trajektorie mit großen rotatorischen Bewegungen um die Hochachse und um die Rollachse vorgegeben wird. Dann bewirkt eine Übergewichtung der Aktuatoren von Achse 7 und 8, dass der Hexapod nur kleine rotatorische Bewegungen ausführt und somit nahe an der Neutralstellung bleibt. So können eventuell auftretende translatorische oder schnelle rotatorische Bewegungen durch den Hexapod ausgeführt werden. Für Fall A und Fall B hat dieser Gewichtungsfaktor keinen großen Einfluss.

Der Gewichtungsfaktor „K“ legt fest, wie groß der Einfluss eines Positions- oder Orientierungsfehlers auf die Gelenkbahnplanung ist. Durch Vergrößerung des Parameters werden eventuell auftretende Abweichungen schneller ausgeglichen. Wird der Wert jedoch zu groß gewählt, kann das Verfahren instabil werden. In Abbildung 19 wird gezeigt, wie sich verschiedene Werte des Parameters „K“ auf den Positionsfehler der Trajektorie von Fall B auswirken. Alle restlichen Parameter wurden wie in Abbildung 18 dargestellt gewählt.

Das oberste Diagramm von Abbildung 19 zeigt, dass durch die Wahl eines zu kleinen Wertes für „K“ der Fehler im Gegensatz zum mittleren Diagramm nur langsam abgebaut werden kann und somit größere Abweichungen auftreten. Wird der Parameter zu hoch eingestellt, wie im untersten Diagramm gezeigt, wird das Verfahren instabil und der Positionsfehler erhöht sich.

Mit den restlichen vier Gewichtungparametern kann die Nullraumgeschwindigkeit q_0 eingestellt werden. Diese ermöglicht durch die richtige Parametrierung, dass Positions- und Geschwindigkeitslimits vermieden werden. Die Nullraumgeschwindigkeit ist von den Parametern „W“ und „K“ abhängig, weshalb die Parameter durch Versuch und Irrtum eingestellt werden müssen. Zwei der vier Parameter („eps_pos“ und „eps_vel“) dienen zum Festlegen des Abstands vom Positions- und Geschwindigkeitslimit, ab dem ein Aktuator durch die Nullraumgeschwindigkeit beeinflusst werden soll. Die anderen beiden Gewichtungparameter („alpha_pos“ und „alpha_vel“) ermöglichen eine Anpassung der Amplitude der Nullraumgeschwindigkeit. Durch eine zu hohe Amplitude kann es dazu kommen, dass das System zu Schwingen beginnt und durch eine zu niedrige, dass die Nullraumgeschwindigkeit ihren Einfluss verliert. Werden die Parameter richtig eingestellt, kann die Nullraumgeschwindigkeit dafür sorgen, dass Positions- und Geschwindigkeitslimits eingehalten werden.

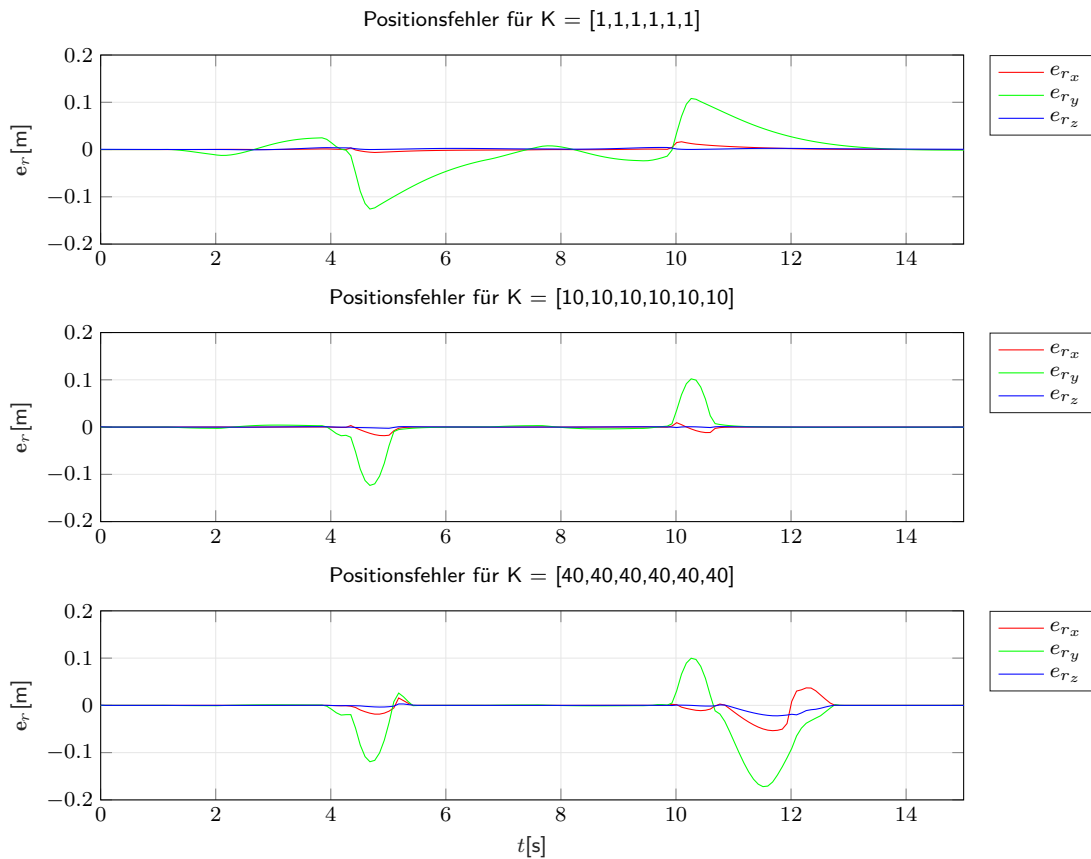


Abbildung 19 Positionsfelder für Fall B bei unterschiedlichen Werten des Gewichtungsparmeters K .

6.3. Resultierende Simulatorbewegungen und Analyse

Fall A

Wie schon erwähnt, wird für Fall A eine kartesische Trajektorie verwendet, welche sich vollständig im Arbeitsraum der Bewegungsplattform darstellen lässt. In Abbildung 20 sind die tatsächliche und die vorgegebene Endeffektorposition und Endeffektororientierung, sowie der Positions- und Orientierungsfehler für Fall A dargestellt. Die Orientierung des Endeffektors wird mithilfe der Kardan-Winkel φ (X-Y-Z) beschrieben.

Bei genauerer Betrachtung der Diagramme in Abbildung 20 ist festzustellen, dass für Fall A die tatsächliche Endeffektorposition \mathbf{r} und Endeffektororientierung φ mit der vorgegebenen Endeffektorposition \mathbf{r}_d und Endeffektororientierung φ_d gut übereinstimmt. Das bedeutet, dass die abgefahrne Trajektorie nur einen kleinen Positionsfehler e_r und einen kleinen Orientierungsfehler e_φ aufweist. Auch im Winkelgeschwindigkeits-, Geschwindigkeits- und Beschleunigungsverlauf, die in Abbildung 24 im Anhang B dargestellt sind, treten keine großen Abweichungen auf. Somit lässt sich festhalten, dass die Simulation der Gelenkbahnplanung für den Fall, dass keine Arbeitsraumgrenzen überschritten werden, funktioniert und für die Bewegungssimulation eingesetzt werden kann. Dieses Ergebnis muss noch an der realen 8-Achs Bewegungsplattform validiert werden, was als nächster Schritt durchgeführt werden sollte.

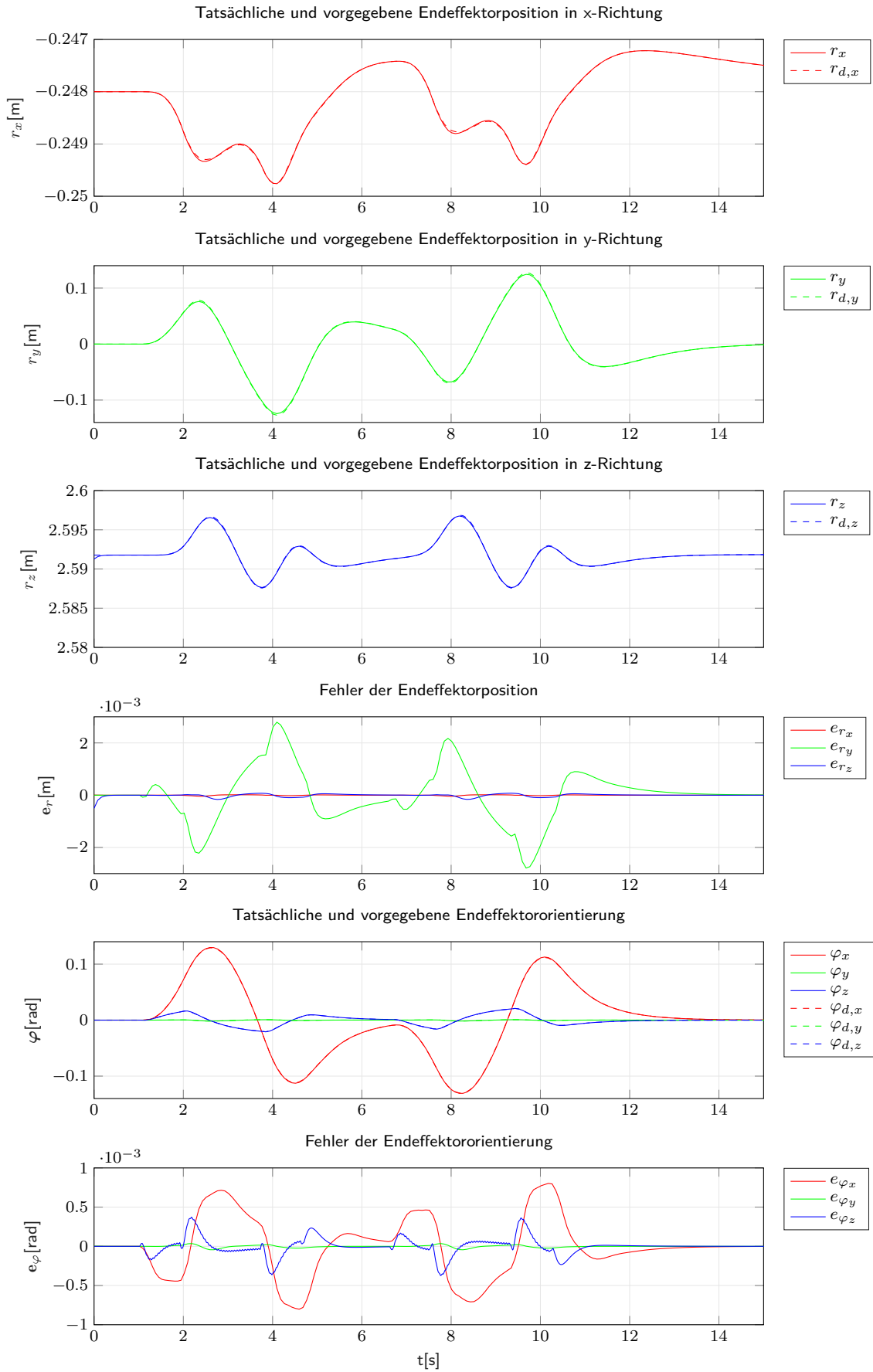


Abbildung 20 Tatsächliche und vorgegebene Endeffektorposition und Endeffektororientierung für Fall A.

Fall B

Durch Fall B wird überprüft, ob die Aktuatorbeschränkungen des Gelenkbahnplanungsmoduls eingehalten werden und wie sich diese auf die Trajektorie des Endeffektors auswirken. Dafür werden die in Abbildung 21 für Fall B dargestellten Verläufe der tatsächlichen und vorgegebenen Endeffektorposition und Endeffektororientierung, sowie der Positions- und Orientierungsfehler untersucht. Die dargestellten Verläufe weisen im Zeitraum zwischen 4s und 6s, sowie zwischen 9s und 11s einen größeren Fehler auf, als im restlichen Zeitbereich. Das liegt daran, dass der Bewegungsplattform in diesen Zeiträumen eine zu große translatorische Bewegung entlang der y-Achse vorgegeben wird, weshalb Aktuatorbeschränkungen aktiviert werden. Diese Beschränkungen können in der Endeffektorposition in y-Richtung aus Abbildung 21 und auch im Verlauf der Aktuatorpositionen, welche in Abbildung 26 in Anhang B dargestellt ist, klar erkannt werden. Die Aktuatoren des Hexapods werden wie in Abbildung 26 dargestellt bei einer Position von ungefähr 0.6m gestoppt. Für die Endeffektorpositionen in x-Richtung und in z-Richtung sowie für die Endeffektororientierung (Abbildung 21) entsteht auch eine Abweichung von der Solltrajektorie, da der Hexapod nur mehr vier Aktuatoren nutzen kann. Durch Vergleich des Positionsfehlers und des Orientierungsfehlers fällt auf, dass der Orientierungsfehler relativ klein ist, was auf die höhere Priorisierung der Orientierung in der Gelenkbahnplanung zurückzuführen ist.

Wie in Abschnitt 4.3 bereits erwähnt, besteht ein Problem der Gelenkbahnplanung darin, dass durch die Positionsbeschränkung der Aktuatoren Beschleunigungssprünge entstehen. Diese Sprünge sind im Beschleunigungsverlauf in Abbildung 26 im Anhang B dargestellt. Darin kann man erkennen, dass die Sprünge beim Aktivieren und Deaktivieren der Beschränkungen entstehen und dass diese, wie in der Parametrierung angegeben, auf $3,5\text{m s}^{-2}$ beschränkt sind. Außerdem sind in Abbildung 26 die Geschwindigkeitsverläufe der Aktuatoren abgebildet, worin zwischen 3s und 5s klar die Geschwindigkeitsbeschränkung bei 0.45m s^{-1} zu erkennen ist.

Die Beschleunigungssprünge der Aktuatoren bewirken auch Beschleunigungssprünge in der kartesischen Trajektorie, welche im Beschleunigungsverlauf in Abbildung 25 im Anhang B dargestellt sind. Dort sind auch die Geschwindigkeits- und Winkelgeschwindigkeitsverläufe für Fall B abgebildet.

Für den Einsatz in der Bewegungssimulation stellen die Sprünge in der Beschleunigung ein Problem dar, da diese höchstwahrscheinlich spürbar sind und folglich ein falsches Bewegungsgefühl entsteht. Zur Lösung könnte versucht werden, die Sprünge auf eine niedrigere Beschleunigung zu begrenzen, zu Filtern oder Aktivierungsfunktionen einzusetzen [36]. Um festzustellen wie problematisch diese Sprünge sind, muss die Gelenkbahnplanung auf der realen 8-Achs Bewegungsplattform getestet werden. Dies wird als nächster Schritt gesehen.

Zu den Zeiträumen in denen keine Aktuatorbeschränkungen auftreten, kann ähnlich wie für Fall A eine gute Einhaltung der vorgegebenen Trajektorie festgestellt werden. Auch nach einer größeren Abweichung von der Solltrajektorie findet die Gelenkbahnplanung schnell zur vorgegebenen Trajektorie zurück, wenn die Aktuatorbeschränkungen aufgehoben werden.

Zusammenfassend kann festgehalten werden, dass die Gelenkbahnplanung gut funktioniert, wenn der Arbeitsraum eingehalten wird. Es ist jedoch davon auszugehen, dass diese erst in der Bewegungssimulation eingesetzt werden kann, wenn eine Lösung zur Beseitigung bzw. Verringerung der Beschleunigungssprünge bei der Verletzung des Arbeitsraums gefunden wurde.

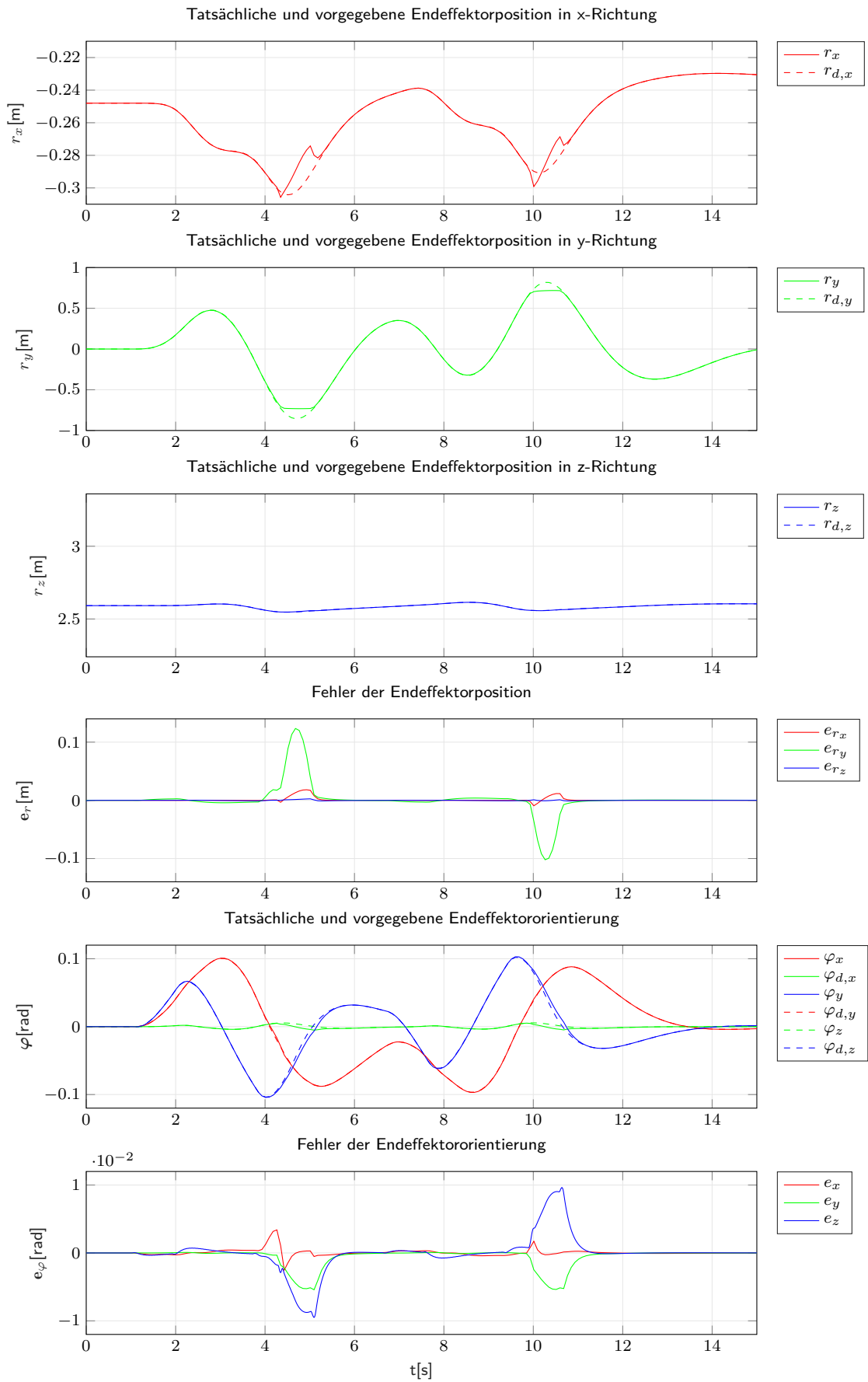


Abbildung 21 Tatsächliche und vorgegebene Endeffektorposition und Endeffektororientierung für Fall B.

7. Antriebsauslegung

In diesem Kapitel wird auf die Auslegung der Antriebe von Achse 7 und Achse 8 näher eingegangen. Dafür werden zu Beginn die benötigten Anforderungen definiert und anschließend in Abschnitt 7.1 die maximal auftretenden Antriebsmomente der beiden Achsen berechnet.

Die zur Auslegung der rotatorischen Antriebe relevanten Größen sind die maximale Drehzahl und das maximale Drehmoment. Das maximale Drehmoment ergibt sich aus der maximalen Drehbeschleunigung. Im Folgenden werden die in dieser Arbeit verwendeten maximalen Drehzahlen und Drehbeschleunigungen von Achse 7 und Achse 8 definiert.

Mit der siebten Achse sollen Drehzahlen erreicht werden, welche ähnlich zu den Drehzahlen von Kurzarm-Humanzentrifugen sind, damit die Bewegungsplattform in Zukunft vielseitig eingesetzt werden kann. Um in der Fahrsimulation übliche Manöver zu simulieren, muss auch die Drehbeschleunigung entsprechend gewählt werden. Wie groß diese sein soll, wurde aus einer Veröffentlichung von Olivari u. a. [40] übernommen.

Durch die achte Achse kann die Gondel um die Rollachse gedreht werden. Damit soll es möglich sein, Rollraten und Rollbeschleunigungen zu erreichen, die im Bereich eines Kampfflugzeugs liegen.

In Tabelle 3 sind die in dieser Arbeit verwendeten maximalen Drehgeschwindigkeiten und Drehbeschleunigungen für Achse 7 und Achse 8 dargestellt.

Tabelle 3 Maximale Drehgeschwindigkeit und Drehbeschleunigung von Achse 7 und Achse 8.

	Drehgeschwindigkeit	Drehbeschleunigung
Achse 7	$230 \text{ }^\circ \text{ s}^{-1}$	$100 \text{ }^\circ \text{ s}^{-2}$
Achse 8	$130 \text{ }^\circ \text{ s}^{-1}$	$130 \text{ }^\circ \text{ s}^{-2}$

7.1. Berechnung der maximalen Antriebsmomente

Zu Beginn wurden manuelle Manöver mit dem Mehrkörpersimulationsmodell der 8-Achs Bewegungsplattform durchgeführt. Dabei wurde versucht, Lastfälle zu finden, bei denen die Momente um Achse 7 und Achse 8 maximal werden. Jedoch stellt dies eine herausfordernde Aufgabe dar, weil die dafür nötigen Bewegungen des Hexapods unter Berücksichtigung der Positions-, Geschwindigkeits- und Beschleunigungslimits der Aktuatoren nicht einfach zu identifizieren sind. Aus diesem Grund wurde zur Berechnung der maximalen Antriebsmomente jeweils folgendes Optimierungsproblem definiert:

$$\begin{aligned}
& \max_{\mathbf{q}} \quad \mathbf{M}_{7/8}(\mathbf{q}(t)) \\
& \text{u.d.N.} \quad \mathbf{q}(0) = \mathbf{q}(t_{end}) = \mathbf{q}_{neutral}, \\
& \quad \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}(t_{end}) = \ddot{\mathbf{q}}(0) = \ddot{\mathbf{q}}(t_{end}) = \mathbf{0}, \\
& \quad s_{min} \leq q_i(t) \leq s_{max}, \\
& \quad |\dot{\mathbf{q}}(t)| \leq \dot{\mathbf{q}}_{max} \quad |\ddot{\mathbf{q}}(t)| \leq \ddot{\mathbf{q}}_{max}, \\
& \quad \text{mit } i = 1, \dots, 6 \quad \text{und } t \in [0, t_{end}].
\end{aligned} \tag{7.1}$$

Darin stellt die zu maximierende Funktion $\mathbf{M}_{7/8}(\mathbf{q}(t))$ das von den Minimalkoordinaten abhängige Antriebsmoment von Achse 7 oder von Achse 8 dar, je nachdem für welche Achse das maximale Antriebsmoment berechnet werden soll. Durch $\mathbf{q}_{neutral}$ wird die Ausgangskonfiguration der Bewegungsplattform beschrieben, worin sich der Hexapod in seiner Neutralstellung befindet. Die beiden zusätzlichen Achsen starten für die Optimierung von Achse 7 in der Nullposition, wie in Abbildung 5 dargestellt. Bei der Optimierung von Achse 8 startet diese um 90° verdreht, da dadurch ein besseres Ergebnis erzielt werden konnte. Ansonsten ist die Startkonfiguration gleich wie in der Optimierung für Achse 7. Die Positionslimits der Aktuatoren werden mit s_{min} und s_{max} bezeichnet und die Geschwindigkeits- und Beschleunigungslimits mit $\dot{\mathbf{q}}_{max}$ und $\ddot{\mathbf{q}}_{max}$. Diese Limits wurden aus den Maschinendaten des Hexapods übernommen und für die beiden zusätzlichen Achsen wurden die in Tabelle 3 definierten Werte verwendet.

Zur Lösung des Optimierungsproblems wurde die DLR Optimization Bibliothek¹ eingesetzt. Mit dieser besteht die Möglichkeit eine sogenannte Trajektorienoptimierung in Verbindung mit einem Modelica Mehrkörpermodell durchzuführen [45]. Dafür muss das Modell Eingänge besitzen, mit denen die Position, Geschwindigkeit oder Beschleunigung der Gelenke vorgegeben werden kann und Ausgänge, welche die für die Optimierung notwendigen Kriterien darstellen. In dieser Arbeit wurde dafür das Mehrkörpersimulationsmodell der 8-Achs Bewegungsplattform aus Abschnitt 5.1 verwendet.

Bei der Trajektorienoptimierung wird für jeden Eingang des Modells eine Trajektorie erzeugt, welche aus N äquidistante Stützstellen besteht, die durch B-Splines k -ten Grades miteinander verbunden sind. Diese Stützstellen werden dann durch einen Optimierungsalgorithmus variiert, sodass ein vorgegebenes Kriterium unter Berücksichtigung der angegebenen Nebenbedingungen minimiert wird. Als Optimierungsalgorithmus wird in dieser Arbeit der SQP-Algorithmus verwendet. Die Abkürzung SQP steht für „sequential quadratic programming“, wobei es sich um ein iteratives Verfahren handelt, mit dem beschränkte nichtlineare Optimierungsprobleme gelöst werden können. Bei der Lösung handelt es sich um ein lokales Optimum. Für eine genauere Erklärung des Verfahrens sei auf [29] verwiesen.

Mit dem SQP-Algorithmus ist es möglich, dass durch Vorgabe einer großen Anzahl an Stützstellen und schlechten Startwerten keine Lösung gefunden wird. Dem kann entgegengewirkt werden, indem zu Beginn eine geringe Anzahl an Stützstellen festgelegt wird, welche anschließend iterativ erhöht wird. Als Startwerte werden dann die in der Iteration zuvor berechneten Trajektorien verwendet.

¹ <https://www.systemcontrolinnovationlab.de/the-dlr-optimization-library/>.

Dabei wird die Anzahl der Stützstellen so lange erhöht, bis keine bessere Lösung mehr gefunden wird. Im Zuge dieser Arbeit konnten mit der beschriebenen Vorgehensweise gute Ergebnisse erzielt werden.

Durch die Optimization Bibliothek werden die Startwerte nach Erhöhung der Stützstellen automatisch aus den Ergebnissen der letzten Iteration berechnet, jedoch kann dabei höchstens ein B-Spline dritten Grades verwendet werden. Durch die Wahl eines B-Splines dritten Grades ist die Trajektorie zweifach stetig differenzierbar. Für die Gelenke des verwendeten Mehrkörpersimulationsmodells, dem die Gelenkposition als Eingang vorgegeben wird, ergibt sich dadurch ein Beschleunigungsverlauf, der aus stückweise linearen Teilen besteht. Dadurch entstehen Sprünge im Ruck, was problematisch für die Umsetzung der generierten Gelenktrajektorien am realen Hexapod sein kann. Aus diesem Grund wurde ein weiteres Optimierungsproblem erstellt, worin die zu maximierenden Antriebsmomente nicht von der Gelenkposition \mathbf{q} , sondern von der Gelenkgeschwindigkeit $\dot{\mathbf{q}}$ abhängen, wodurch ruckfreie Gelenktrajektorien entstehen. Das Mehrkörpersimulationsmodell musste dafür so abgeändert werden, dass die Geschwindigkeiten der Gelenke als Eingänge vorgegeben werden können. Das Optimierungsproblem ist daher wie folgt definiert:

$$\begin{aligned} \max_{\dot{\mathbf{q}}} \quad & \mathbf{M}_{7/8}(\dot{\mathbf{q}}(t)) \\ \text{u.d.N.} \quad & \mathbf{q}(0) = \mathbf{q}_{neutral}, \\ & \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}(t_{end}) = \ddot{\mathbf{q}}(0) = \ddot{\mathbf{q}}(t_{end}) = \mathbf{0}, \\ & s_{min} \leq q_i(t) \leq s_{max}, \\ & |\dot{\mathbf{q}}(t)| \leq \dot{\mathbf{q}}_{max} \quad |\ddot{\mathbf{q}}(t)| \leq \ddot{\mathbf{q}}_{max}, \\ & \text{mit } i = 1, \dots, 6 \quad \text{und } t \in [0, t_{end}]. \end{aligned} \tag{7.2}$$

Ergeben sich für beide Optimierungsprobleme Antriebsmomente, welche nur gering voneinander abweichen, so kann man davon ausgehen, dass die Gelenktrajektorien auch am realen Hexapod umgesetzt werden können. Das bedeutet, dass auch die berechneten maximalen Antriebsmomente erreicht werden können. Dies muss jedoch in zukünftigen Schritten validiert werden.

Für alle Optimierungsprobleme wurden zu Beginn sechs Stützstellen festgelegt, welche dann so lange iterativ erhöht wurden, bis keine bessere Lösung mehr gefunden wurde. Für t_{end} wurde bei allen Lösungen 3 s eingestellt, außer für die Lösung des ersten Optimierungsproblems für Achse 7, hier wurde fünf Sekunden eingestellt, da auf diese Weise ein besseres Ergebnis erzielt werden konnte. Für den SQP-Algorithmus ist es wichtig, dass eine genaue Lösung berechnet wird, weshalb die Toleranz der Integration auf 10^{-12} gesetzt wurde. Die Toleranz der Lösung des SQP-Algorithmus wurde mit 10^{-6} festgelegt. Um die Rechenzeit zu verkürzen, wurde außerdem die parallele Berechnung auf acht Kernen aktiviert.

In Abbildung 22 sind die Verläufe der Antriebsmomente aus den Lösungen der beiden Optimierungsprobleme für Achse 7 dargestellt. Wie zu sehen ist, stimmen die maximalen Antriebsmomente mit einer geringen Abweichung überein, weshalb angenommen wird, dass die ermittelten Gelenktrajektorien auf dem realen Hexapod umgesetzt werden können. In Abbildung 27 und 28 in Anhang C sind außerdem die Positions-, Geschwindigkeits- und Beschleunigungsverläufe der Aktuatoren dar-

gestellt, welche aus der Lösung der Optimierungsprobleme resultieren. Darin ist zu erkennen, dass die Aktuatoren in den meisten Fällen ihre Geschwindigkeits- und Beschleunigungslimits erreichen, weshalb es sich mit großer Wahrscheinlichkeit um ein globales Optimum handelt.

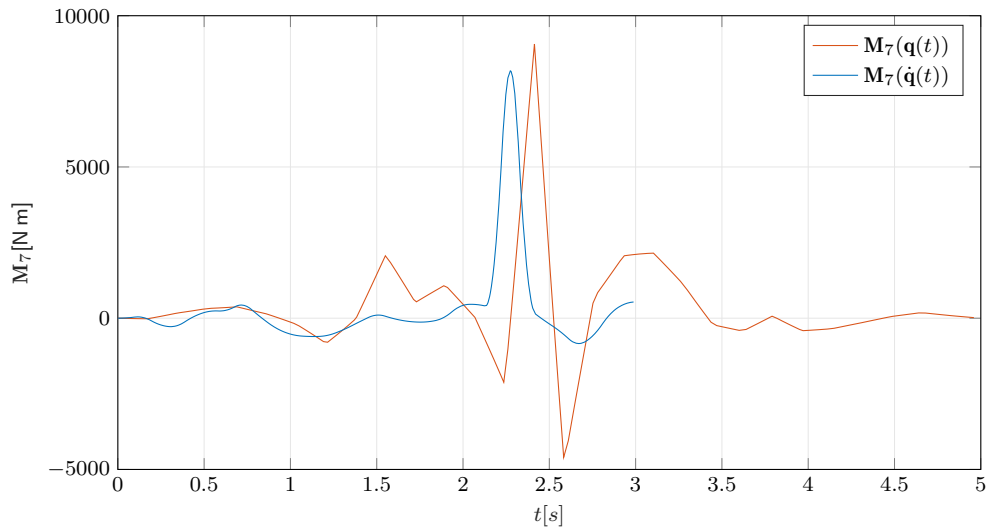


Abbildung 22 Verläufe der Antriebsmomente von Achse 7, welche aus den beiden Optimierungsproblemen berechnet wurden.

Weiter sind in Abbildung 23 die Verläufe der Antriebsmomente für Achse 8 dargestellt, die aus der Lösung der Optimierungsprobleme ermittelt wurden. Auch hier stimmen die Maximalwerte überein, weshalb die ermittelten Gelenktrajektorien mit großer Wahrscheinlichkeit auf dem Hexapod umgesetzt werden können. Des Weiteren sind in Abbildung 29 und 30 in Anhang C die Positions-, Geschwindigkeits- und Beschleunigungsverläufe der Aktuatoren für die Lösungen der Optimierungsprobleme von Achse 8 abgebildet. Wie in den Lösungen von Achse 7, erreichen die Aktuatoren auch hier zu großen Teilen ihre Geschwindigkeits- und Beschleunigungslimits. Es kann daher davon ausgegangen werden, dass es sich dabei höchstwahrscheinlich um ein globales Optimum handelt.

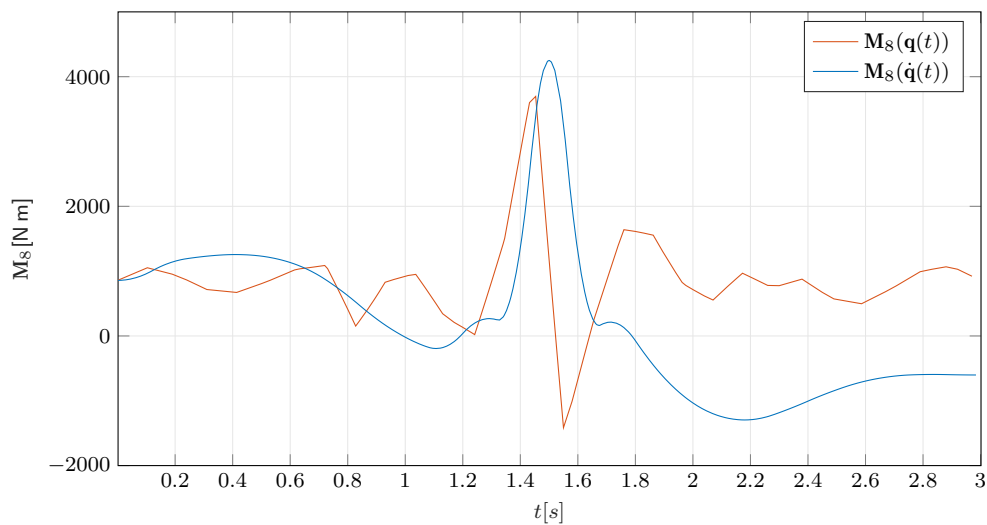


Abbildung 23 Verläufe der Antriebsmomente von Achse 8, welche aus den beiden Optimierungsproblemen berechnet wurden.

Die maximalen Antriebsmomente von Achse 7 und Achse 8 ergeben sich zu:

- $M_7 = 9065\text{N m}$
- $M_8 = 4250\text{N m}$

Zusammenfassen kann festgehalten werden, dass mit der beschriebenen Vorgehensweise die maximalen Antriebsmomente für Achse 7 und Achse 8 ermittelt werden konnten. Es ist jedoch darauf hinzuweisen, dass diese Berechnungen wiederholt werden müssen, wenn die Bauweise der Bewegungsplattform final festgelegt wurde. Weiter müssen die ermittelten Ergebnisse in einem nächsten Schritt auf dem realen Hexapod validiert werden.

Da sich die Antriebsmomente noch verändern können, wird an dieser Stelle verzichtet, einen geeigneten Antriebsmotor vorzustellen.

8. Zusammenfassung und Ausblick

Gegenstand der vorliegenden Arbeit ist die Entwicklung eines Gelenkbahnplanungsmoduls für die 8-Achs Bewegungsplattform und die Antriebsauslegung für die beiden zusätzlichen Achsen.

Zu Beginn der Arbeit wurden existierende Bewegungsplattformen und deren Steueralgorithmen recherchiert sowie die 8-Achs Bewegungsplattform präsentiert. Die größten Vorteile der neuartigen Bewegungsplattform stellt die hohe Nutzlast und die steife Bauweise dar, wodurch hochdynamische Bewegungen ermöglicht werden. Zusätzlich gibt es im Arbeitsraum der Bewegungsplattform keine Singularitäten, was die Erstellung der Gelenkbahnplanung vereinfacht und robuster macht.

Anschließend wurden die Grundlagen zur Berechnung der Kinematik dargelegt, welche für die Erstellung der Gelenkbahnplanung benötigt werden. Aufgrund der hybriden Struktur und der Redundanz des Systems wurde die inverse Kinematik auf Geschwindigkeitsebene berechnet. Dafür wurde eine Methode verwendet, mit der es möglich ist, mehrere Aufgaben mit verschiedenen Prioritätslevels zu definieren und Positionsbeschränkungen direkt in die inverse Kinematik zu integrieren. Um Positionsfehler auszugleichen, wurde eine inverse Kinematik mit Rückführung vorgestellt.

Aufbauend auf der inversen Kinematik wurde die Gelenkbahnplanung erstellt, mit der unter Berücksichtigung der Positions-, Geschwindigkeits- und Beschleunigungslimits der Aktuatoren kartesische Trajektorien auf der Bewegungsplattform abgefahren werden können. Diese wurde mithilfe der Modellierungs- und Simulationsumgebung Dymola implementiert, womit auch ein Mehrkörpersimulationsmodell und eine 3D-Visualisierung erstellt wurden.

Um die Funktionsweise der Gelenkbahnplanung zu verifizieren, wurde das Mehrkörpersimulationsmodell der Bewegungsplattform verwendet. Aus der Verifizierung ging hervor, dass eine vorgegebene kartesische Trajektorie mit Hilfe der entwickelten Gelenkbahnplanung richtig abgefahren werden kann, wenn keine Aktuatorlimits verletzt werden. Wichtig dafür ist die richtige Parametrierung, welche zu großen Teilen durch Versuch und Irrtum bestimmt wurde.

Wird eine kartesische Trajektorie vorgegeben, bei der ein Aktuator ein Positions-, Geschwindigkeits- oder Beschleunigungslimit erreicht, konnte gezeigt werden, dass diese Limits vollständig eingehalten werden. Ein Problem der Gelenkbahnplanung ergibt sich durch die Integration der Positionsbeschränkung in die inverse Kinematik. Dadurch entstehen Geschwindigkeitssprünge in den Aktuatoren, wenn eine Beschränkung aktiviert oder deaktiviert wird, welche mit einer Beschleunigungsbeschränkung begrenzt wurden. Diese begrenzten Sprünge sind höchstwahrscheinlich spürbar und sorgen folglich für ein falsches Bewegungsgefühl. Um die richtige Parametrierung zu ermitteln, welche dafür sorgt, dass die Sprünge möglichst unbemerkt bleiben, muss die Gelenkbahnplanung an der realen Bewegungsplattform getestet bzw. eingestellt werden.

Weiter folgt zum Ende dieser Arbeit die Bestimmung der Lastfälle, bei denen die maximalen Antriebsmomente für Achse 7 und Achse 8 auftreten. Dafür wurde ein Optimierungsproblem definiert, mit dem die Antriebsmomente maximiert werden. Um dieses zu lösen, wurde die DLR Optimization Bibliothek in Verbindung mit dem Mehrkörpersimulationsmodell der Bewegungsplattform eingesetzt, womit Gelenktrajektorien ermittelt wurden, welche dafür sorgen, dass die Antriebsmomente maxi-

mal werden. Als Optimierungsalgorithmus wurde der SQP-Algorithmus verwendet, der ein lokales Optimum bestimmt. Da durch die ermittelten Gelenktrajektorien der Großteil der Aktuatoren die Geschwindigkeits- und Beschleunigungslimits erreicht, wird davon ausgegangen, dass die berechnete Lösung ein globales Optimum darstellt.

Die resultierenden maximalen Antriebsmomente können genutzt werden, um die Antriebsmotoren der beiden zusätzlichen Achsen zu bestimmen. Da die Struktur der 8-Achs Bewegungsplattform noch nicht fertig ausgelegt ist und sich deshalb die Bauweise noch verändern kann, muss das Optimierungsproblem erneut durchgeführt werden, wenn die Auslegung abgeschlossen ist. Da sich die Antriebsmomente noch verändern können, wurde in dieser Arbeit darauf verzichtet, einen geeigneten Antriebsmotor vorzustellen.

Ausblick

Um das Gelenkbahnplanungsmodul für den Einsatz in der Bewegungssimulation zu optimieren, soll in den nächsten Schritten versucht werden, die Geschwindigkeitssprünge auf eine niedrige Beschleunigung zu begrenzen, zu filtern oder mit Aktivierungsfunktionen einen glatten Übergang zu erzeugen. Welche Methode sich am besten eignet, muss durch Tests an der realen 8-Achs Bewegungsplattform ermittelt werden. Zusätzlich sollte validiert werden, ob eine gewünschte kartesische Trajektorie auch am realen System eingehalten wird, wie in der Verifizierung festgestellt.

Weiter ist festzuhalten, dass die maximalen Antriebsmomente mit dem idealisierten Mehrkörpersimulationsmodell ermittelt wurden, weshalb die zugehörigen Gelenktrajektorien am realen System auf ihre Abbildbarkeit überprüft werden sollten.

Literatur

- [1] A. Hamish Jamson, Anthony J. Horrobin und Robin A. Auckland. "Whatever Happened to the LADS? Design and development of the new University of Leeds Driving Simulator". In: *Driving Simulation Conference North America 2007*. 2007.
- [2] Alain Liégeois. "Automatic Supervisory Control of the Configuration and Behavior of Multi-body Mechanisms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 7.12 (1977), S. 868–871. ISSN: 0018-9472. DOI: 10.1109/TSMC.1977.4309644.
- [3] Gianluca Antonelli, Filippo Arrichiello und Stefano Chiaverini. "The NSB control: a behavior-based approach for multi-robot systems". In: *Paladyn, Journal of Behavioral Robotics* 1.1 (2010), S. 48–56. ISSN: 2081-4836. DOI: 10.2478/s13230-010-0006-0. URL: <https://www.degruyter.com/document/doi/10.2478/s13230-010-0006-0/html>.
- [4] Gianluca Antonelli, Stefano Chiaverini und Giuseppe Fusco. "Kinematic Control of Redundant Manipulators with On-Line End-Effector Path Tracking Capability Under Velocity and Acceleration Constraints". In: *IFAC Proceedings Volumes* 33.27 (2000), S. 183–188. ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)37926-0.
- [5] Tobias Bellmann. "Optimierungsbasierte Bahnplanung für interaktive robotische Bewegungssimulatoren". Dissertation. Universität der Bundeswehr München. URL: <https://elib.dlr.de/93031/>.
- [6] Tobias Bellmann, Johann Heindl, Matthias Hellerer, Richard Kuchar, Karan Sharma und Gerd Hirzinger. "The DLR Robot Motion Simulator Part I: Design and setup". In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, S. 4694–4701. ISBN: 978-1-61284-386-5. DOI: 10.1109/ICRA.2011.5979913.
- [7] Willem Bles und Eric Groen. "The DESDEMONA Motion Facility: Applications for Space Research". In: *Microgravity Science and Technology* 21.4 (2009), S. 281–286. ISSN: 1875-0494. DOI: 10.1007/s12217-009-9120-1.
- [8] Ilian Bonev. *The True Origins of Parallel Robots*. 2003. URL: <http://www.parallemic.org/Reviews/Review007.html> (besucht am 18.04.2021).
- [9] Bosch Rexroth AG, Hrsg. *eMotion-2700*. URL: <https://www.boschrexroth.com/en/xc/industries/machinery-applications-and-engineering/motion-simulation-technology/products-and-solutions/6dof-motion-platform/emotion-2700/index> (besucht am 18.04.2021).
- [10] S. Chiaverini. "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators". In: *IEEE Transactions on Robotics and Automation* 13.3 (1997), S. 398–410. ISSN: 1042296X. DOI: 10.1109/70.585902.
- [11] Florent Colombet, Zhou Fang und Andras Kemeny. "Pitch tilt rendering for an 8-DOF driving simulator". In: *Driving Simulation Conference 2015*. 2015.
- [12] John J. Craig. *Introduction to robotics: Mechanics and control*. Fourth edition. NY NY: Pearson, 2018. ISBN: 9780133489798.

- [13] Arati S. Deo und Ian D. Walker. "Overview of damped least-squares methods for inverse kinematics of robot manipulators". In: *Journal of Intelligent and Robotic Systems* 14.1 (1995), S. 43–68. ISSN: 0921-0296. DOI: 10.1007/BF01254007.
- [14] Peter Deuffhard. *Newton methods for nonlinear problems: Affine invariance and adaptive algorithms*. corr. 2. print. Bd. 35. Springer series in computational mathematics. Berlin: Springer, 2006. ISBN: 3540210997.
- [15] Domagoj Jakobovi. *Forward Kinematics of a Stewart Platform Mechanism*. 2002. URL: https://www.researchgate.net/publication/2564027_Forward_Kinematics_of_a_Stewart_Platform_Mechanism.
- [16] Antônio O. Dourado und C. A. Martin. "New concept of dynamic flight simulator, Part I". In: *Aerospace Science and Technology* 30.1 (2013), S. 79–82. ISSN: 1270-9638. DOI: 10.1016/j.ast.2013.07.005.
- [17] Holger Duda, Sunjoo K. Advani und Mario Potter. "Design of the DLR AVES Research Flight Simulator". In: *AIAA Modeling and Simulation Technologies (MST) Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 8192013. DOI: 10.2514/6.2013-4737.
- [18] Felix Ellensohn. "Urban Motion Cueing Algorithms: Trajectory Optimization for Driving Simulators". Dissertation. TU München, 2020.
- [19] Hilding Elmqvist, Sven Erik Mattsson und Martin Otter. "Modelica - a language for physical system modeling, visualization and interaction". In: *Computer-Aided Design and Test for High-Speed Electronics 1999*. Hrsg. von IEEE. Piscataway: IEEE, Sept. 1999, S. 630–639. ISBN: 0-7803-5500-8. DOI: 10.1109/CACSD.1999.808720.
- [20] Martin Fischer. "Motion-Cueing-Algorithmen für eine realitätsnahe Bewegungssimulation". Dissertation. TU Braunschweig, 2009. DOI: 10.24355/DBBS.084-200910051056-0.
- [21] Flight Physiological Centre, Hrsg. *The Dynamic Flight Simulator (DFS)*. URL: <https://fpc.qinetiq.com/what-we-do/our-facilities/the-dynamic-flight-simulator.aspx> (besucht am 21. 04. 2021).
- [22] Nikhil J.I. Garrett und Matt Best. "Driving simulator motion cueing algorithms: a survey of the state of the art". In: *Proceedings of the 10th International Symposium on Advanced Vehicle Control (AVEC)*. 2010, S. 183–186.
- [23] P. Robuffo Giordano, C. Masone, J. Tesch, M. Breidt, L. Pollini und H. H. Bühlhoff. "A novel framework for closed-loop robotic motion simulation - part I: Inverse kinematics design". In: *IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, Alaska, USA, 3-7 May 2010* (2010), S. 3876–3883. ISSN: 1050-4729. DOI: 10.1109/ROBOT.2010.5509647.
- [24] V. E. Gough und S. G. Whitehall. "Universal tyre test machine". In: *Ninth International Technical Congress F.I.S.I.T.A.* 1962, S. 117–137.
- [25] VI-grade GmbH, Hrsg. *DiM DYNAMIC simulator | VI-grade*. 20.04.2021. URL: <https://www.vi-grade.com/en/products/dim-dynamic-simulator/>.

- [26] Basile Graf. *Quaternions and dynamics*. URL: <https://arxiv.org/pdf/0811.2889>.
- [27] H. Hanafusa, T. Yoshikawa und Y. Nakamura. "Analysis and Control of Articulated Robot Arms with Redundancy". In: *IFAC Proceedings Volumes 14.2* (1981), S. 1927–1932. ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)63754-6.
- [28] Xiguang Huang, Qizheng Liao, Shimin Wei, Xu Qiang und Shuguang Huang. "Forward Kinematics of the 6-6 Stewart Platform with Planar Base and Platform Using Algebraic Elimination". In: *IEEE International Conference on Automation and Logistics, 2007*. Piscataway, NJ: IEEE Service Center, 2007, S. 2655–2659. ISBN: 978-1-4244-1530-4. DOI: 10.1109/ICAL.2007.4339029.
- [29] H. -D Joos. *MOPS - Multi-Objective Parameter Synthesis; User's Guide*. 2002. URL: <https://elib.dlr.de/12170/>.
- [30] Kyoung Kim, Moon Kim, Young Moon und Min Lee. "Application of Vehicle Driving simulator Using New Washout Algorithm and Robust Control". In: *International joint conference SICE-ICASE, 2006*. Piscataway, NJ: IEEE Service Center, 2006, S. 2121–2126. ISBN: 89-950038-4-7. DOI: 10.1109/SICE.2006.315563.
- [31] Rafał Lewkowicz. "A centrifuge-based flight simulator: Optimization of a baseline acceleration profile based on the motion sickness incidence". In: *Acta Astronautica* 164 (2019), S. 23–33. ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2019.07.007. URL: <https://www.sciencedirect.com/science/article/pii/S0094576518317296>.
- [32] Kevin M. Lynch und Frank C. Park. *Modern robotics: Mechanics, planning, and control*. Cambridge, New York, NY und Port Melbourne: Cambridge University Press, 2017. ISBN: 9781107156302.
- [33] Mark Wilkinson, Timothy Leo Brown und Omar Ahmad. "The National Advanced Driving Simulator (NADS) Description and Capabilities in Vision-Related Research". In: *Optometry - Journal of the American Optometric Association* 83.6 (2012), S. 79–84. ISSN: 1529-1839.
- [34] Carlo Masone, Paolo Robuffo Giordano und Heinrich H. Bulthoff. "Mechanical design and control of the new 7-DOF CyberMotion simulator". In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, S. 4935–4942. ISBN: 978-1-61284-386-5. DOI: 10.1109/ICRA.2011.5980436.
- [35] Signe Moe, Gianluca Antonelli, Andrew R. Teel, Kristin Y. Pettersen und Johannes Schrimpf. "Set-Based Tasks within the Singularity-Robust Multiple Task-Priority Inverse Kinematics Framework: General Formulation, Stability Analysis, and Experimental Results". In: *Frontiers in Robotics and AI* 3 (2016), S. 16. ISSN: 2296-9144. DOI: 10.3389/frobt.2016.00016.
- [36] Signe Moe, Jan Tommy Gravdahl und Kristin Y. Pettersen. "Set-Based Control for Autonomous Spray Painting". In: *IEEE Transactions on Automation Science and Engineering* 15.4 (2018), S. 1785–1796. ISSN: 1545-5955. DOI: 10.1109/TASE.2018.2801382.
- [37] Takahiko Murano, Takashi Yonekawa, Masami Aga und Sueharu Nagiri. "Development of High-Performance Driving Simulator". In: *SAE International Journal of Passenger Cars - Mechanical Systems* 2.1 (2009), S. 661–669. ISSN: 1946-4002. DOI: 10.4271/2009-01-0450.

- [38] Yoshihiko Nakamura. *Advanced robotics: Redundancy and optimization*. Addison-Wesley series in electrical and computer engineering Control engineering. Reading, Mass.: Addison-Wesley, 1991. ISBN: 0201151987.
- [39] Yoshihiko Nakamura, Hideo Hanafusa und Tsuneo Yoshikawa. "Task-Priority Based Redundancy Control of Robot Manipulators". In: *The International Journal of Robotics Research* 6.2 (1987), S. 3–15. DOI: 10.1177/027836498700600201.
- [40] M. Olivari, P. Pretto, J. Venrooij und H. H. Bülthoff. "Defining the kinematic requirements for a theoretical driving simulator". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 61 (2019), S. 5–15. ISSN: 13698478. DOI: 10.1016/j.trf.2017.11.006. URL: <https://elib.dlr.de/77222/>.
- [41] Mario Olivari, Frank M. Drop, Mikhail Katliar und Heinrich H. Bulthoff. "Driving Simulators with Hexapod Motion System: Adding a Yaw Turntable". In: *Driving Simulation Conference 2018 Europe VR*. 2018. (Besucht am 20. 04. 2021).
- [42] M. Onuki, Y. Suda, Y. Tamahashi, H. Komine und K. Masushita. "Study on improving angular acceleration feel of the driving simulator and its effects on drivers' behaviors". In: *International Journal of Intelligent Transportation Systems Research*. 2006.
- [43] Martin Otter, Hilding Elmqvist und Sven Erik Mattsson. "The New Modelica MultiBody Library". In: *3rd International Modelica Conference*. Hrsg. von Peter Fritzson. 2003, S. 311–330. URL: <https://elib.dlr.de/11987/>.
- [44] P. Baerlocher und R. Boulic. "Task-priority formulations for the kinematic control of highly redundant articulated structures". In: *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*. 1998, 323–329 vol.1. DOI: 10.1109/IR0S.1998.724639.
- [45] Andreas Pfeiffer. "Optimization Library for Interactive Multi-Criteria Optimization Tasks". In: *9th International Modelica Conference*. 2012, S. 669–679. URL: <https://elib.dlr.de/77222/>.
- [46] Friedrich Pfeiffer und Thorsten Schindler. *Einführung in die Dynamik*. 3. Aufl. Berlin: Springer Vieweg, 2014. ISBN: 9783642410468. DOI: 10.1007/978-3-642-41046-8.
- [47] Jürgen Pitz, Minh-Tri Nguyen, Gerd Baumann und Hans-Christian Reuss. "Combined motion of a hexapod with a xy table system for lateral movements". In: *Driving Simulation Conference 2014*. 2014. (Besucht am 20. 04. 2021).
- [48] L. D. Reid, M. A. Nahon, Transportation Development Centre, University of Toronto. Institute for Aerospace Studies und University of Toronto. Institute of Aerophysics. *Flight Simulation Motion-base Drive Algorithms: Developing and testing the equations*. UTIAS report. Institute for Aerospace Studies, University of Toronto, 1985.
- [49] D.Gordon E. Robertson und James J. Dowling. "Design and responses of Butterworth and critically damped digital filters". In: *Journal of electromyography and kinesiology : official journal of the International Society of Electrophysiological Kinesiology* 13.6 (2003), S. 569–573. ISSN: 1050-6411. DOI: 10.1016/s1050-6411(03)00080-4.

- [50] B. Siciliano und J.-J.E. Slotine. "A general framework for managing multiple tasks in highly redundant robotic systems". In: *Robots in unstructured environments*. New York, N.Y und Piscataway, NJ: IEEE, 1991, 1211–1216 vol.2. ISBN: 0-7803-0078-5. DOI: 10.1109/ICAR.1991.240390.
- [51] Bruno Siciliano. "A closed-loop inverse kinematic scheme for on-line joint-based robot control". In: *Robotica* 8.3 (1990), S. 231–243. ISSN: 02635747. DOI: 10.1017/S0263574700000096.
- [52] Bruno Siciliano, Giuseppe Oriolo, Lorenzo Sciavicco und Luigi Villani. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. London: Springer London, 2009. ISBN: 9781846286421. DOI: 10.1007/978-1-84628-642-1.
- [53] Stefano Chiaverini und Bruno Siciliano. "The Unit Quaternion: A Useful Tool for Inverse Kinematics of Robot Manipulators". In: *Systems Analysis Modelling Simulation* 35.1 (1999), S. 45–60. ISSN: 0232-9298.
- [54] D. Stewart. "A Platform with Six Degrees of Freedom". In: *Proceedings of the Institution of Mechanical Engineers* 180.1 (1965), S. 371–386. ISSN: 0020-3483.
- [55] Reiner Suikat. "The new dynamic driving simulator at DLR". In: *Driving simulation conference 2005*. 2005. URL: https://www.nads-sc.uiowa.edu/dscna/2005/papers/the_new_dynamic_driving_simulator_%20dlr.pdf.
- [56] Daniel Whitney. "Resolved Motion Rate Control of Manipulators and Human Prostheses". In: *IEEE Transactions on Man Machine Systems* 10.2 (1969), S. 47–53. ISSN: 0536-1540. DOI: 10.1109/TMMS.1969.299896.

A. Elementardrehungen und Einheitsvektoren

Die Elementardrehungen um die einzelnen kartesischen Koordinatenachsen sind folgendermaßen definiert:

$$\mathbf{A}_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{pmatrix},$$
$$\mathbf{A}_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{pmatrix},$$
$$\mathbf{A}_z(\gamma) = \begin{pmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Ein Einheitsvektor ist ein Vektor der Länge Eins. Für den dreidimensionalen Vektorraum \mathbb{R}^3 sind diese wie folgt definiert:

$$\mathbf{e}_x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{e}_y = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{e}_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

B. Weitere Simulationsergebnisse der Gelenkbahnplanung

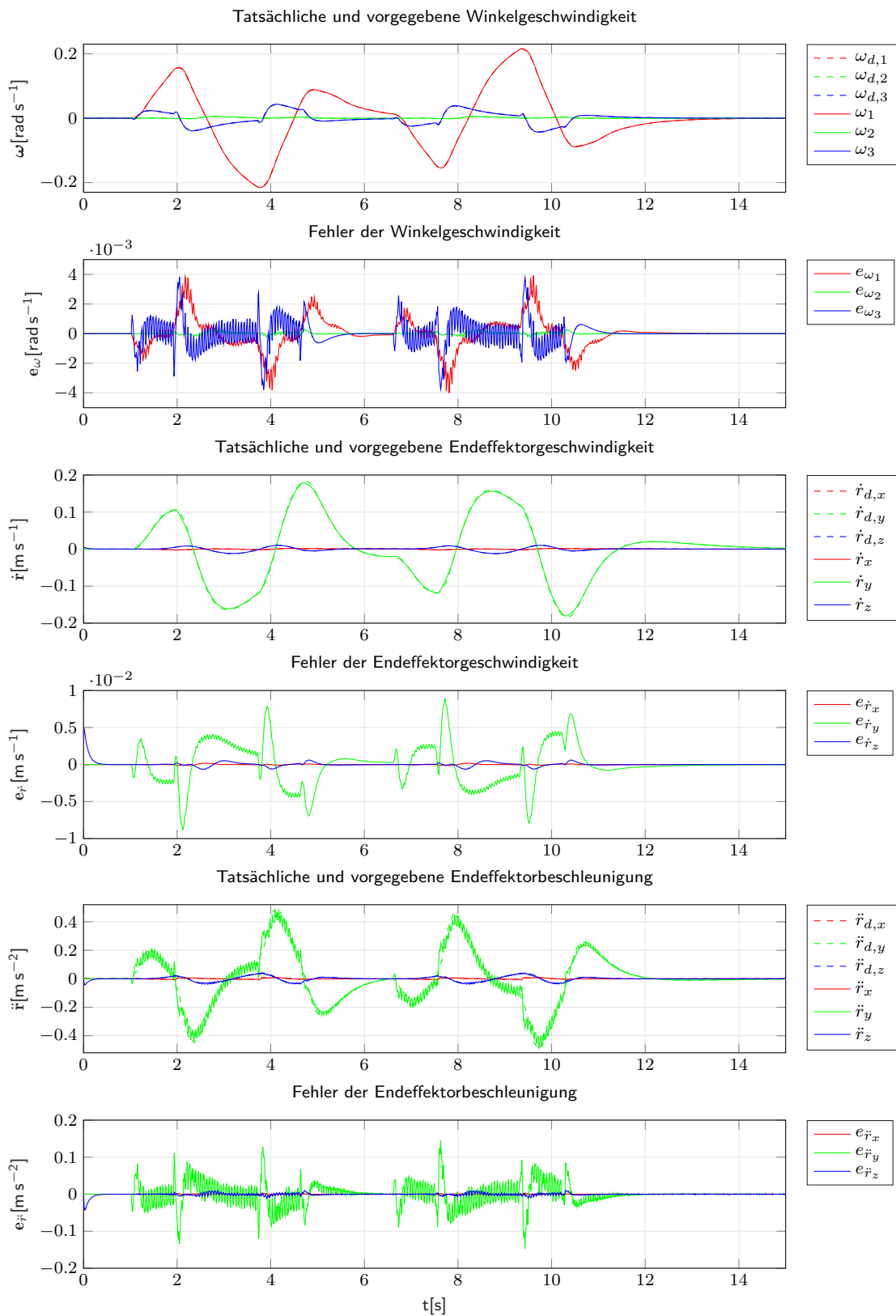


Abbildung 24 Tatsächliche und vorgegebene Winkelgeschwindigkeit, Geschwindigkeit und Beschleunigung des Endeffektors für Fall A.

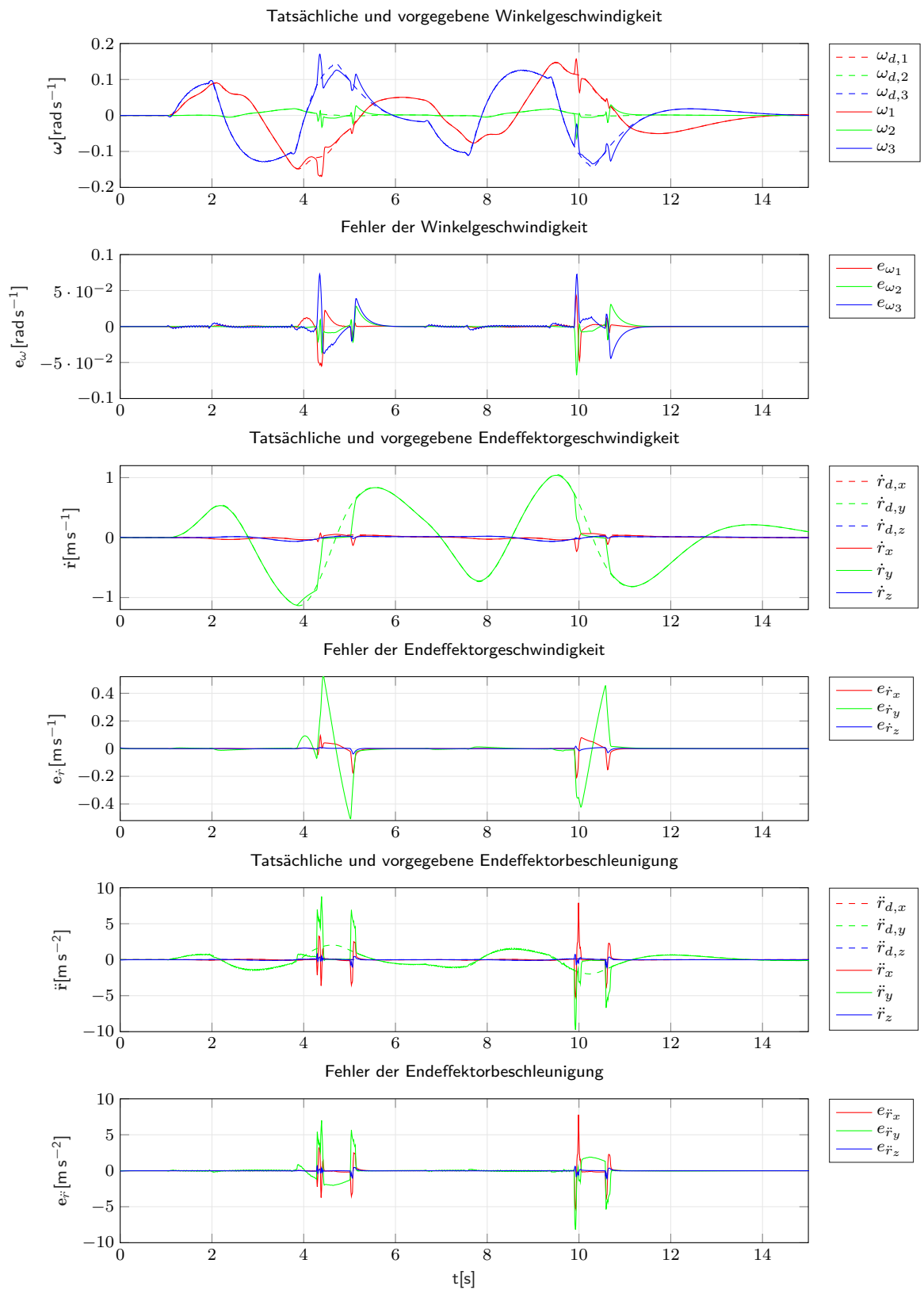


Abbildung 25 Tatsächliche und vorgegebene Winkelgeschwindigkeit, Geschwindigkeit und Beschleunigung des Endeffektors für Fall B.

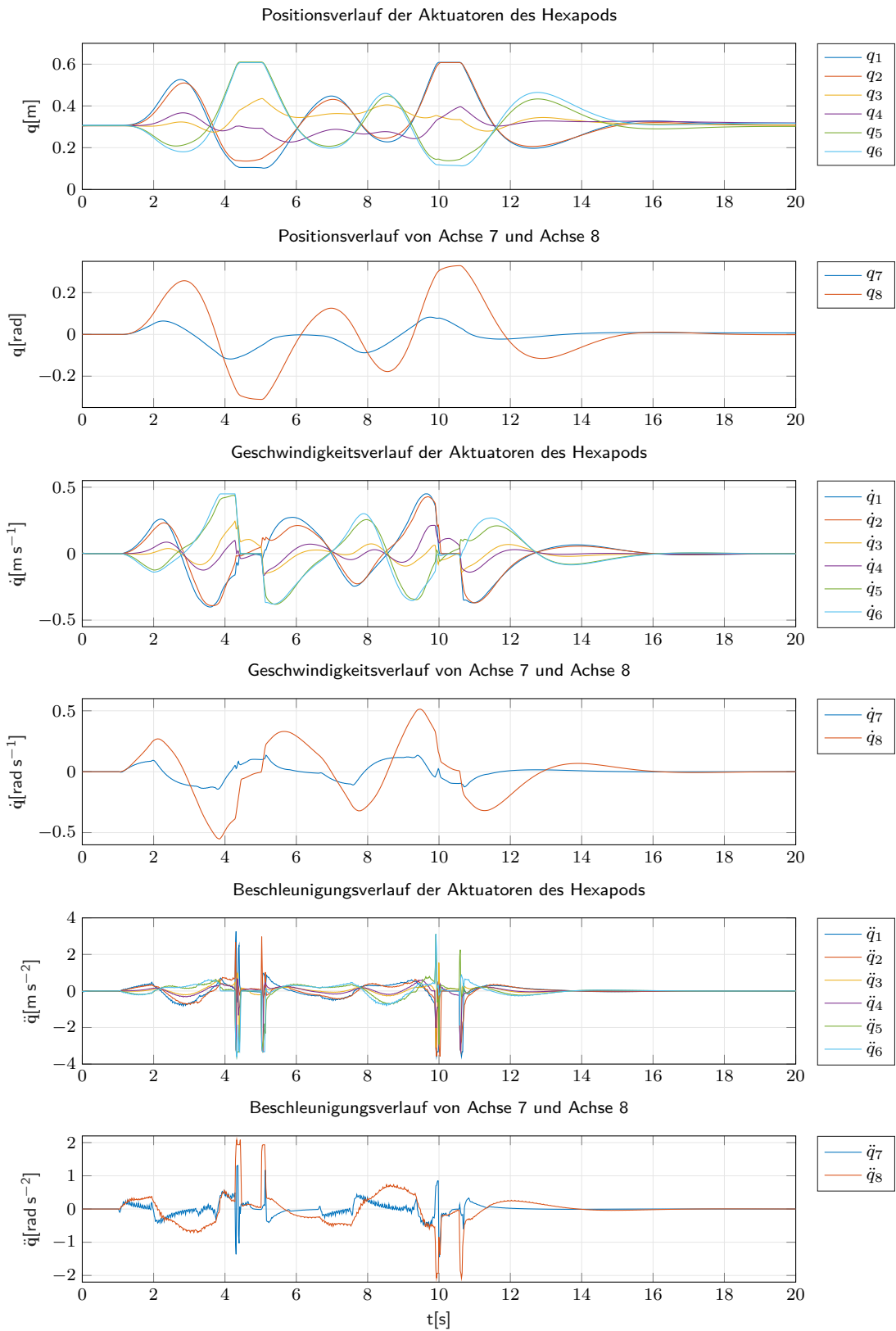


Abbildung 26 Positionsverlauf, Geschwindigkeitsverlauf und Beschleunigungsverlauf der Aktuatoren der 8-Achs Bewegungsplattform für Fall B.

C. Weitere Simulationsergebnisse der Antriebsauslegung

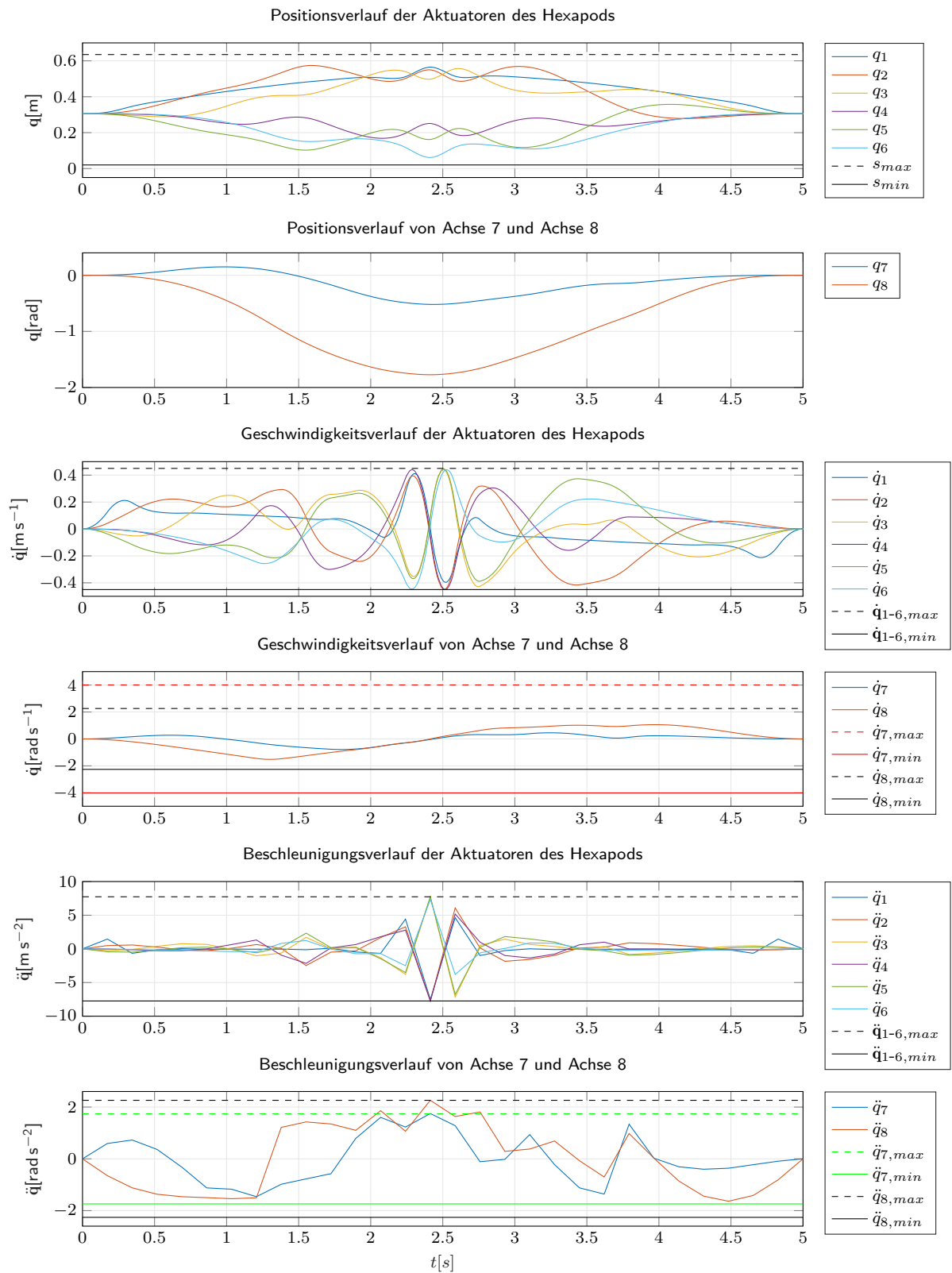


Abbildung 27 Positions-, Geschwindigkeits- und Beschleunigungsverläufe der Aktuatoren der 8-Achs Bewegungsplattform, welche sich durch Lösen des Optimierungsproblems (7.1) für Achse 7 ergeben.

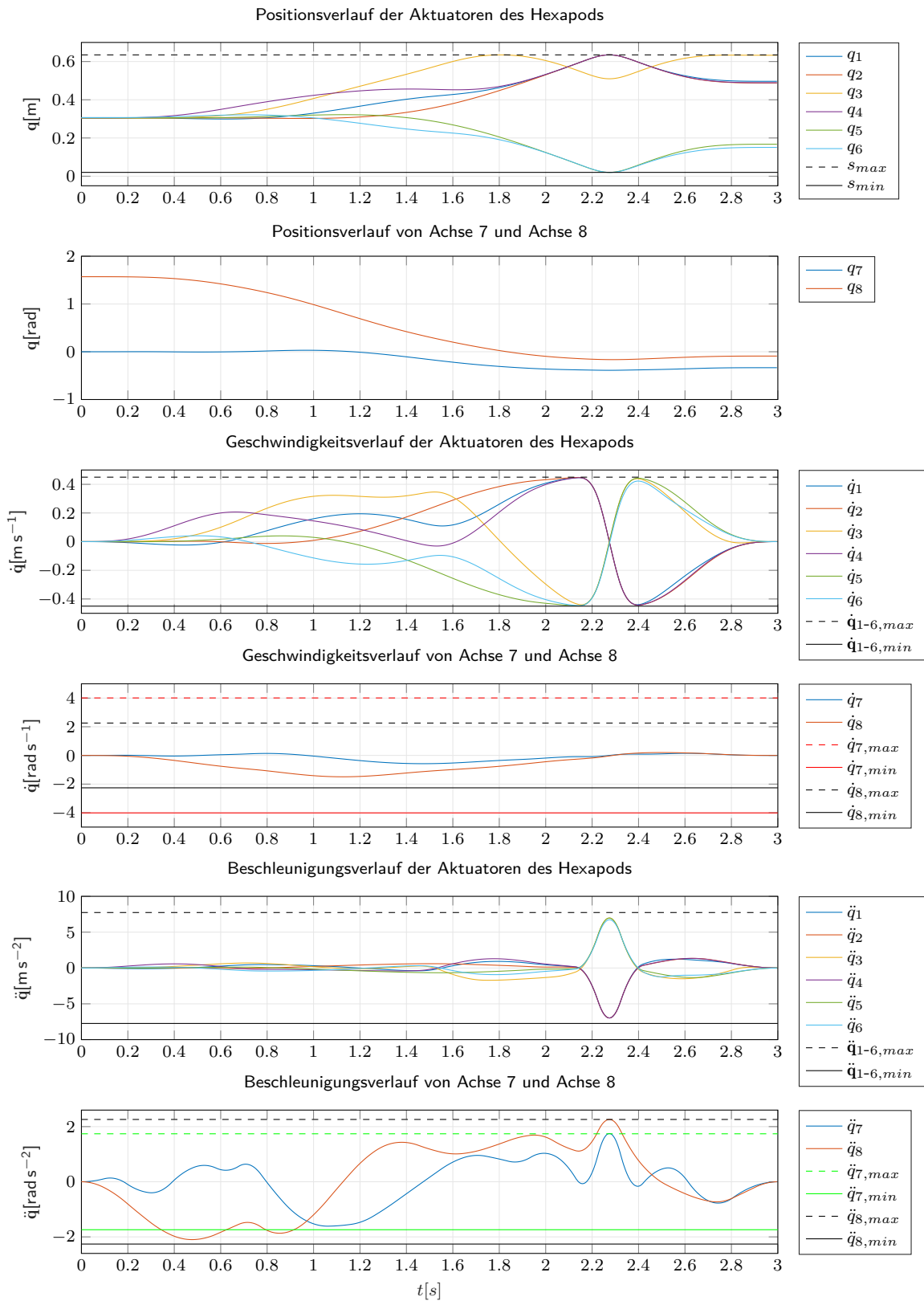


Abbildung 28 Positions-, Geschwindigkeits- und Beschleunigungsverläufe der Aktuatoren der 8-Achs Bewegungsplattform, welche sich durch Lösen des Optimierungsproblems (7.2) für Achse 7 ergeben.

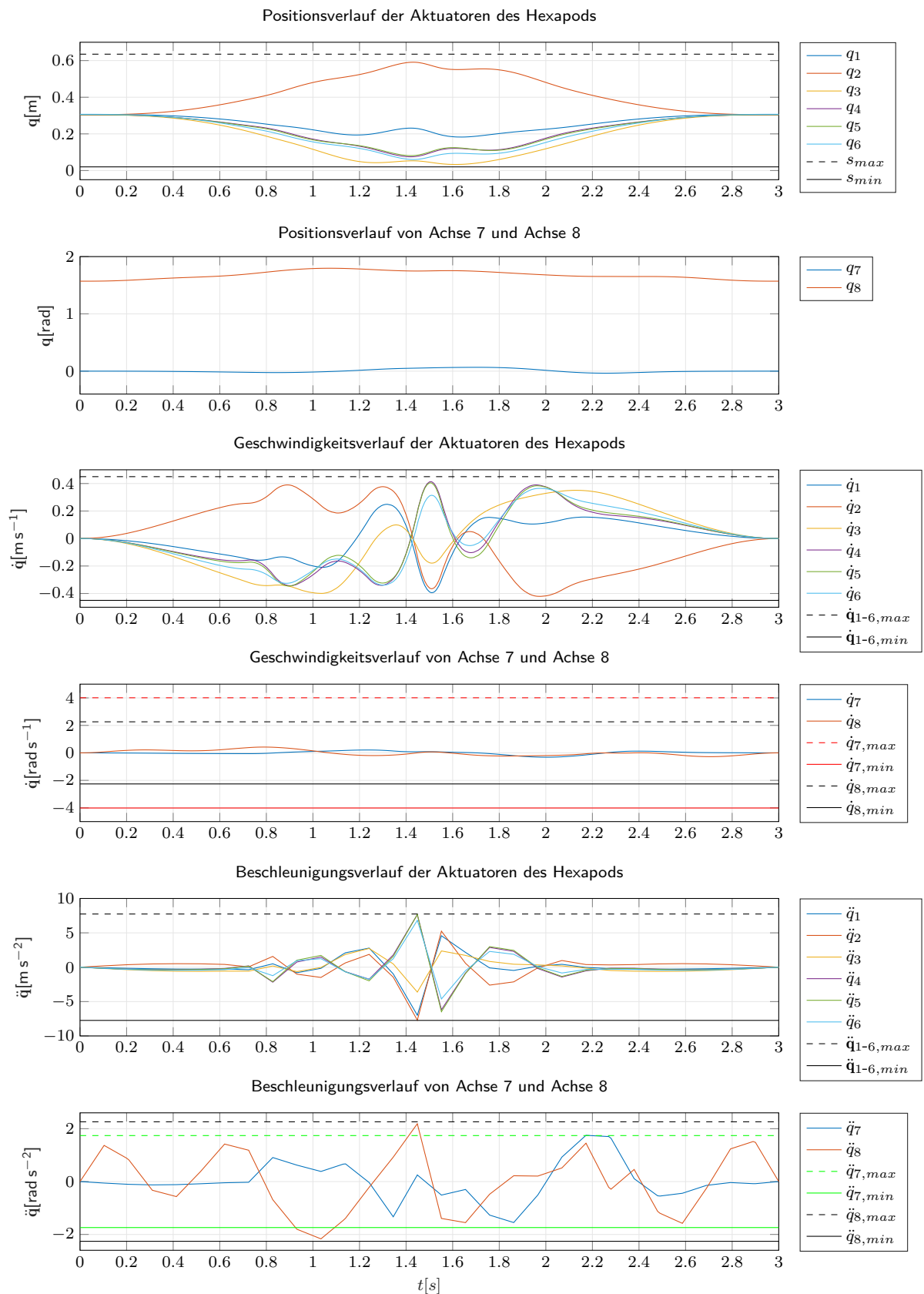


Abbildung 29 Positions-, Geschwindigkeits- und Beschleunigungsverläufe der Aktuatoren der 8-Achs Bewegungsplattform, welche sich durch Lösen des Optimierungsproblems (7.1) für Achse 8 ergeben.

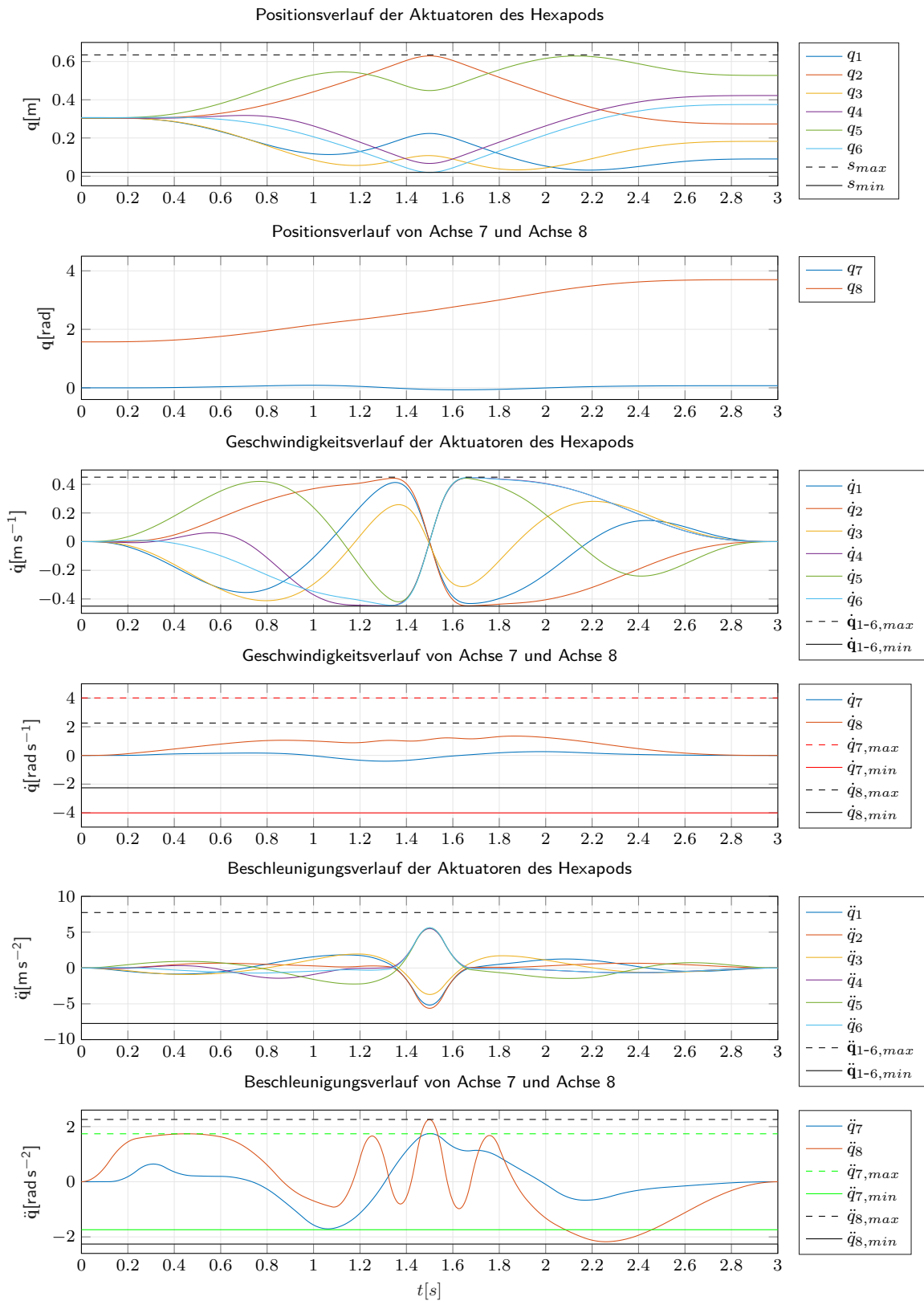


Abbildung 30 Positions-, Geschwindigkeits- und Beschleunigungsverläufe der Aktuatoren der 8-Achs Bewegungsplattform, welche sich durch Lösen des Optimierungsproblems (7.2) für Achse 8 ergeben.

Erklärung

Ich versichere hiermit, dass ich die von mir eingereichte Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort, Datum, Unterschrift